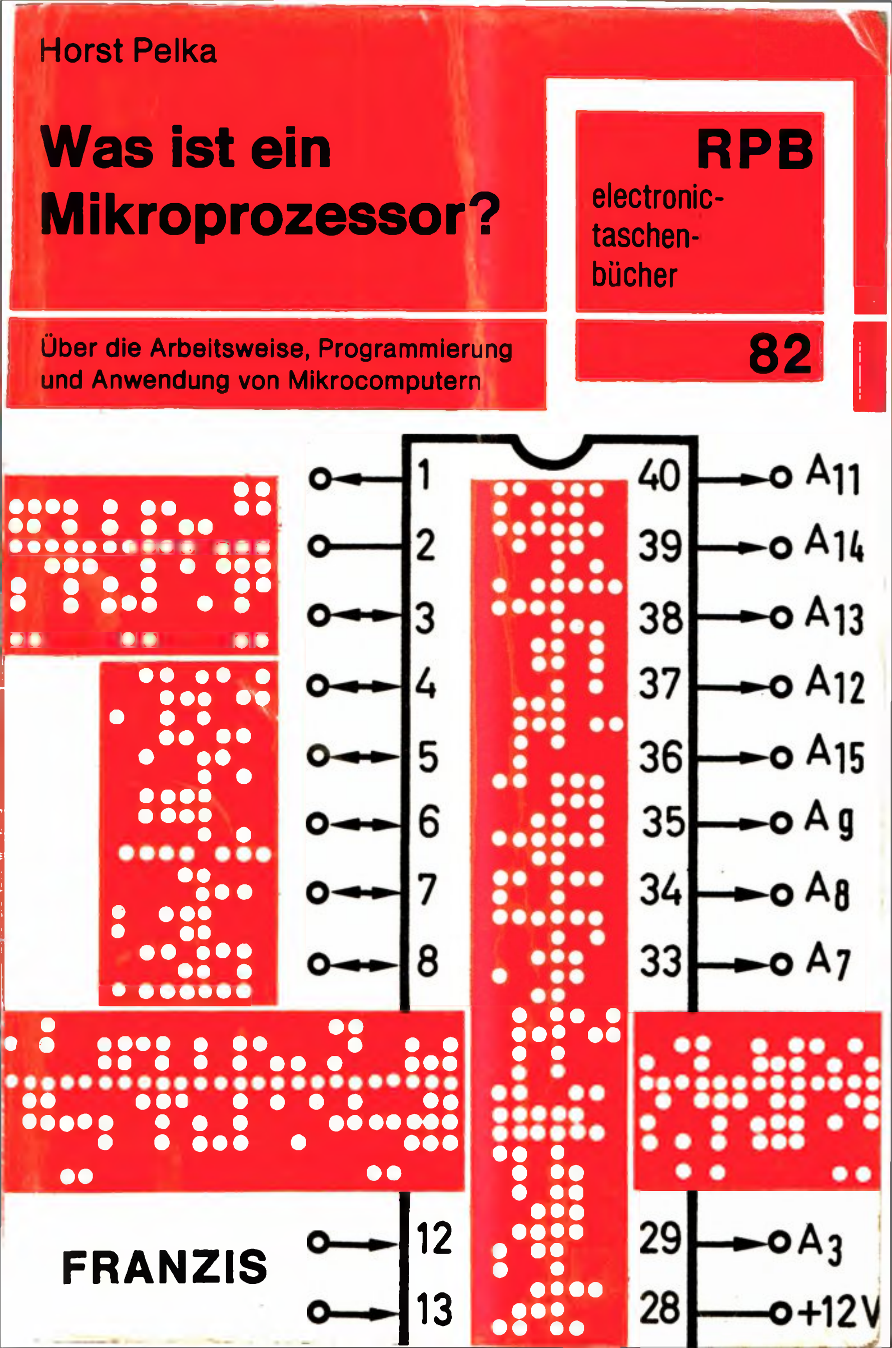

Horst Pelka
Was ist ein Mikroprozessor?
RPB
electronic-taschen-bücher
Über die Arbeitsweise, Programmierung und Anwendung von Mikrocomputern
82
1
2
3
4
5
6
7
8
40
39
38
37
36
35
34
33
A11
A14
A13
A12
A15
A9
A8
A7
12
13
29
28
A3
+12V
FRANZIS

RPB electronic-taschenbücher Titelverzeichnis

Jeder Einfach-Band umfaßt etwa 60–70 Seiten, enthält bis zu 70 Bilder und Tabellen. Mehrfach-Bände entsprechend. Die Nummer vor dem Titel ist die Bestell-Nummer. Preisänderungen und Liefermöglichkeiten vorbehalten.

Preise: Einfachband (●) DM 4.80; Doppelband (● ●) DM 7.80; Dreifachband (● ● ●) DM 9.80; Vierfachband (● ● ● ●) DM 12.80.

4 ● ● Der Hobby-Elektroniker lernt messen (Nührmann)

5 ● ● Brand- und Einbruchmeldeanlagen (Bartels).

6 ● Antennen für Rundfunk- und Fernseh-Empfang (Mende). 15. Aufl.

7 ● Niederfrequenz-Verstärker mit Röhren und Transistoren (Kühne). 14. Aufl.

8 ● ● ● ABC der Elektronik-Orgel (Wuschek)

9 ● ● Tonbandgeräte-Praxis (Junghans). 12. Aufl.

11 ● ● ● Mikrofone (Kühne). 9. Aufl.

16 ● ● ● Widerstandskunde für Elektroniker (Hoffmeister). 7. Aufl.

19 ● ● ● Optoelektronik (Ratheiser/Pichler).

20 ● ● Methodische Fehlersuche in Rundfunkempfängern (Renardy/Lummer). 14. Aufl.

22 ● ● ● Lehrgang Radiotechnik Band I (Jacobs). 12. Aufl.

24 ● ● ● desgl., Band II (Jacobs). 8. Aufl.

25 ● ● ● Amateur-Funkfernschreibtechnik RTTY (Pietsch).

29 ● ● ● Kleines ABC der Elektroakustik (Büscher). 7. Aufl.

30 ● ● ● UHF-Amateurfunk-Antennen (Reithofer)

34 ● ● ● Von der Mengenlehre zur Schaltalgebra (Siegfried).

35 ● ● Systemanalyse und System-Design (Lewandowski).

38 ● ● ● ● SSB- und ISB-Technik (Pelka).

40 ● Fachwörter der Elektronik (Franz). 2. Aufl.

44 ● ● ● KW- und UKW-Amateurfunk-Antennen (Diefenbach). 10. Aufl.

46 ● Sendertabelle (Schneider/Redlich). 2. Aufl.

50 ● ● Prakt. Antennenbau (Mende). 16. Aufl.

51 ● ● Kleine Fernseh-Bildfehler-Fibel (Gies/Kirsch). 2. Aufl.

53 ● ● ● Elektroflug (Bruß).

54 ● ● ● Schaltalgebra im Experiment (Kleemann)

55 ● ● ● Fernsehtechnik von A biz Z (Conrad). 5. Aufl.

58 ● ● Morselehrgang (Diefenbach). 10. Aufl.

59 ● ● Funk-Entstörungs-Praxis (Mende). 5. Aufl.

60 ● ● Die Widerstand-Kondensator-Schaltung (Schneider). 7. Aufl.

64 ● ● Einführung in die Operationsverstärker-Technik (Hirschmann). 2. Aufl.

65 ● ● ● Operationsverstärker-Anwendung (Hirschmann).

68 ● ● ● Kleine Elektronik-Formelsammlung (Rose). 14. Aufl.

69 ● ● Integrierte Schaltungen in Frage und Antwort (Hibberd).

70 ● ● ● Laser, Maser, Hologramme (Ferretti)

72 ● ● Drahtlose Fernsteuerung von Flugmodellen (Schultheiss). 6. Aufl.

73 ● ● Wie liest man eine Schaltung? (Benda).

80 ● ● ● ● Das Spulenbuch – Hochfrequenzspulen (Sutaner). 6. Aufl.

82 ● ● Was ist ein Mikroprozessor? (Pelka). 3. Aufl.

83 ● ● Mikrofon-Aufnahmetechnik (Praetzel/Warnke)

84 ● ● Fernsehantennen-Praxis (Mende). 13. Aufl.

85 ● ● ● HiFi-Schaltungs- und Baubuch (Kühne/Horst). 7. Aufl.

87 ● ● Methodische Fehlersuche in der Industrie-Elektronik (Benda).

88 ● ● Schliche und Kniffe für Radiopraktiker (Kühne). 7. Aufl.

90 ● Netzgeräte mit ICs (Sehrig).

92 ● ● ● Einfache Experimente mit digitalen Schaltgliedern (Wirsum).

Horst Pelka

Was ist ein Mikroprozessor?

Über die Arbeitsweise, Programmierung und Anwendung von Mikrocomputern

Mit 41 Abbildungen und 5 Tabellen

3., neu bearbeitete Auflage



FRANZIS-VERLAG MÜNCHEN

Nr. 82 der RPB-electronic-taschenbücher

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Pelka, Horst
Was ist ein Mikroprozessor? : Über d. Arbeitsweise, Programmierung u. Anwendung von Mikrocomputern. – 3., neu bearb. Aufl. – München : Franzis-Verlag, 1978.
(|RPB-Elektronik-Taschenbücher|; RPB-electronic-taschenbücher; Nr. 82)
ISBN 3-7723-0823-6

1978

Franzis-Verlag GmbH, München

Druck: Franzis-Druck GmbH, Karlstr. 35 München 2
Printed in Germany. Imprimé en Allemagne.

ISBN 3-7723-0823-6

Vorwort

Was ist ein Mikroprozessor? Der Mikroprozessor ist nach der Erfindung des Transistors und der Planartechnik eine der wichtigsten Erfindungen der Elektronik.

Der dringende Bedarf an universellen LSI-Standard-Schaltungen führte zur Entwicklung des Mikroprozessors. Die Halbleiterindustrie lernte, wie man tausend und abertausend Transistoren auf einem winzigen Siliziumkristall integrieren konnte, stand aber vor dem großen Problem, daß diese hochintegrierten Schaltungen nur so spezialisiert eingesetzt werden konnten, daß die viel zu geringe Stückzahl einer Rationalisierung und Kostensenkung im Wege stand. Es galt also eine hochintegrierte Schaltung in Standard-Ausführung für eine breite Anwendung zu finden. Das Ergebnis ist der Mikroprozessor. Er reagiert auf Eingangssignale und erzeugt Ausgangssignale, welche von einer Folge von Befehlen bestimmt werden. Sein Verhalten ist also nicht anders als das eines fest verdrahteten Logiknetzwerkes aus vielen TTL-Bausteinen. Nur kann man durch Ändern der Programme das logische Verhalten schnell neuen Bedürfnissen anpassen, was man bei einem festen Netzwerk aus Logikbausteinen nicht konnte. Wenn man nun das Programm in einem Festwertspeicher unterbringt, ist es möglich, die logische Funktion durch Austausch dieses einen Bausteines zu verändern.

Man sieht also, daß man den Mikroprozessor nicht als Gebilde allein betrachten darf, sondern besser in einem System zusammen mit den zugehörigen Speichern sowie Ein- und Ausgangsbausteinen. Deshalb wurden zum besseren Verständnis zunächst im Abschnitt 3 zwei Mikrocomputersysteme (Systeme mit Mikroprozessoren) besprochen, ehe im Abschnitt 4 auf den Mikroprozessor selbst eingegangen wird.

Die wichtigsten Vorteile von Mikroprozessor-Systemen sind niedrige Kosten, weniger Bauteile, größere Zuverlässigkeit und außergewöhnliche Vielseitigkeit.

Die Typen 8008 und 8080 wurden als Beispiele gewählt. Beides sind Standard-Typen, die auf dem freien Markt eine große Verbreitung gefunden haben. Der 8008 ist eine sehr ko-

stengünstige Lösung mit nur einer Sammelschiene (Bus), während der 8080 die zweite Generation mit getrennten Daten- und Adressensammelschienen verkörpert.

Auf andere, am Markt befindliche Typen wird anschließend eingegangen.

Da dem Elektroniker viele Fachausdrücke aus der Computertechnik nicht so geläufig sind, befindet sich am Ende des Buches ein Glossar zum Nachschlagen.

Vorwort zur 3. Auflage

In der 3. Auflage wurde der behandelte Stoff um die neuen Mikroprozessoren 8085 und 8048 ergänzt. Der Umfang der Literaturhinweise, auf den neuesten Stand mit über 300 Titeln, sprengte den Rahmen eines RPB-Zweifachbandes. Die Literaturhinweise wurden daher herausgenommen. Sie finden sie in dem kürzlich erschienenen Buch "Von der Schaltalgebra zum Mikroprozessor".

Horst Pelka

Wichtiger Hinweis

Die in diesem Buch wiedergegebenen Schaltungen und Verfahren werden ohne Rücksicht auf die Patentlage mitgeteilt. Sie sind ausschließlich für Amateur- und Lehrzwecke bestimmt und dürfen nicht gewerblich genutzt werden*).

Alle Schaltungen und technischen Angaben in diesem Buch wurden vom Autor mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Trotzdem sind Fehler nicht ganz auszuschließen. Der Verlag sieht sich deshalb gezwungen, darauf hinzuweisen, daß er weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen kann. Für die Mitteilung eventueller Fehler sind Autor und Verlag jederzeit dankbar.

*) Bei gewerblicher Nutzung ist vorher die Genehmigung des möglichen Lizenzinhabers einzuholen.

Inhalt

1 Einleitung

Seit einigen Jahren ist der Begriff "Mikroprozessor" in der Fachliteratur aufgetaucht. Was steckt dahinter? Die Frage ist nicht so einfach zu beantworten. Sie muß im Zusammenhang mit der EDV betrachtet werden.

Der Mikroprozessor ist ein Teil eines Mikrocomputers. Zur Komplettierung zu einer funktionsfähigen Einheit werden noch Speicher sowie die Ein- und Ausgabeeinheiten benötigt.

Erschrecken Sie bitte nicht über die vielen Fachausdrücke in den nachfolgenden Abschnitten, eine Erklärung aller in diesem Buch verwendeten Fachausdrücke finden Sie in dem Abschnitt 10 (Fachausdrücke — Glossar).

Um das Kernstück, die "microprogramable central processing unit" — abgekürzt MP oder Mikroprozessor — gruppieren sich die anderen Schaltkreise. Obwohl das Wort "microprogramable", zu deutsch mikroprogrammierbar vorkommt, werden in der Praxis nur einige MP, bei weitem nicht alle, durch ein Mikroprogramm gesteuert.

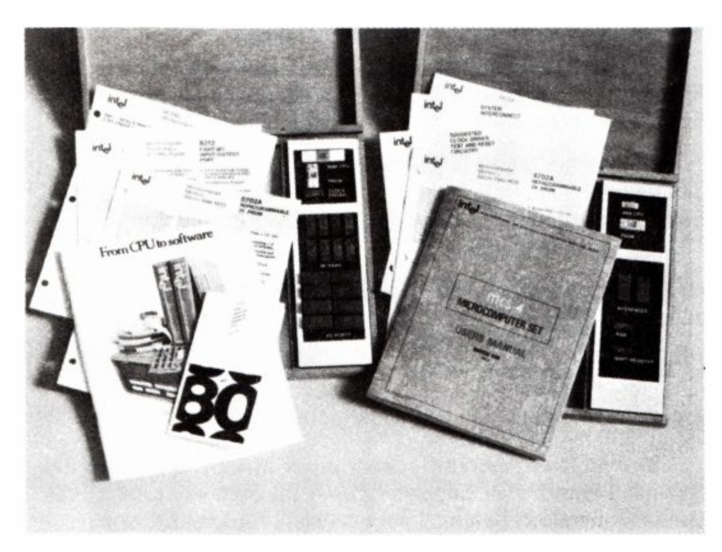
Während in der ersten Generation die Mikroprozessoren in P-Kanal-Technik eine Kapazität von 4 Bit besaßen, erreicht die zweite Generation bereits 8 Bit und kann binäre und alphanumerische Daten verarbeiten. Ein Einsatzgebiet ist in intelligenten Terminals, die mit EBCDI- oder ASCII-Code arbeiten.

Die zweite Generation ist im Gegensatz zur ersten Generation im N-Kanal-MOS-Technologie ausgeführt und daher im Datenfluß wesentlich schneller. Moderne 16-Bit-Chips können 70 bis 90 Befehle mit einer Zyklus-Zeit von 1 bis 2 µs verarbeiten.

Mikroprozessoren können nur aus einem LSI-IS bestehen. Es gibt aber auch Vorschläge, solche Prozessoren mit Standard-Bausteinen in MSI-Technik aufzubauen. Es werden für einfache Konfigurationen etwa 12 Bausteine gebraucht.

Ein Mikrocomputer setzt sich nun aus dem Mikroprozessor, dem Speicher sowie Ein- und Ausgabeeinheiten zusammen. *)
Die verschiedenen Programme werden durch Austausch des Lesespeichers PROM (programable read only memory) realisiert. Im Gegensatz zur fest verdrahteten CPU, bei der viele Vorgänge parallel ablaufen, ist die Verarbeitung in einem Mikroprozessor seriell. Er benötigt daher längere Rechenzeiten.

Zur besseren Einführung in die Technik der Mikroprozessoren soll daher im nachfolgenden Kapitel zunächst einmal der grundsätzliche Aufbau einer EDV besprochen werden.



1.1 Intels erstes Mikrocomputersystem 4004 und das neue System 8080. Aufnahme: Intel Corporation

---

*)*Aus Abb. 1.1* (Mitte) ist ersichtlich, wie zu dem Mikroprozessor (8080 CPU) durch Hinzufügung weiterer Bausteine ein Mikrocomputer zusammengestellt werden kann.

2 EDV-Grundlagen

Eine EDV-Anlage besteht aus einer Zentraleinheit und den peripheren Einheiten (*Abb. 2.1*). In der Zentraleinheit sind das Rechenwerk, das Steuerwerk, der Arbeitsspeicher und verschiedene Register zusammengefaßt. Zur Peripherie gehört die Programm-Bibliothek mit Organisations-Programmen, Übersetzungs-Programmen, Umsetz-Programm, Testhilfen und Anwender-Programmen. Bei der Neueinlesung eines Programmes werden die Lochkarten im Stanzer gestanzt. Über den Kartenleser wird das Primärprogramm eingelesen. Zur Bedienung der Zentraleinheit dient die Bedienungsschreibmaschine. Die Kommunikation von weiter entfernten Terminals geschieht über den Modem. Zur Ausgabe wird ein Drucker oder ein Sichtgerät benutzt.

Der Prozessor ist ein Teil der Zentraleinheit einer Elektronischen Datenverarbeitungsanlage (EDV-Anlage). Er enthält das Rechenwerk und das Leitwerk. Im Rechenwerk werden Rechenoperationen, zu denen neben arithmetischen Operationen auch "Boolesche" Verknüpfungen, verschieben, vergleichen, runden usw. gehören, ausgeführt.

Das Rechenwerk enthält im allgemeinen folgende drei Register, die jedoch bei vielen Anlagen dem Programmierer nicht zugänglich sind:

1. Den Akkumulator als zentrales Register
2. Das Multiplikator-Quotientenregister
3. Das Multiplikandenregister

Das Leitwerk — auch control unit, Steuerwerk oder Kommandowerk genannt,ist einer der wichtigsten, selbständigen Teile der Zentraleinheit. Die Aufgaben des Leitwerkes sind:

1. Steuerung der Reihenfolge, in der die Befehle des Programmes aus dem Arbeitsspeicher gelesen werden,
2. Entschlüsselung der Befehle,
3. Modifikation der Befehle, Änderung der Adressen unter Berücksichtigung der Indexregister, Durchführung von Adressensubstitutionen,

4. Erzeugung von Signalen, die in Rechenwerk, Arbeitsspeicher oder den Eingabe- und Ausgabegeräten die Ausführung der Befehle auslösen,
5. Verarbeitung der Bedingungen bei bedingten Befehlen,
6. Sicherstellung, daß kein Teil eine Aufgabe erhält, solange er noch nicht mit einer früheren Aufgabe fertig ist.

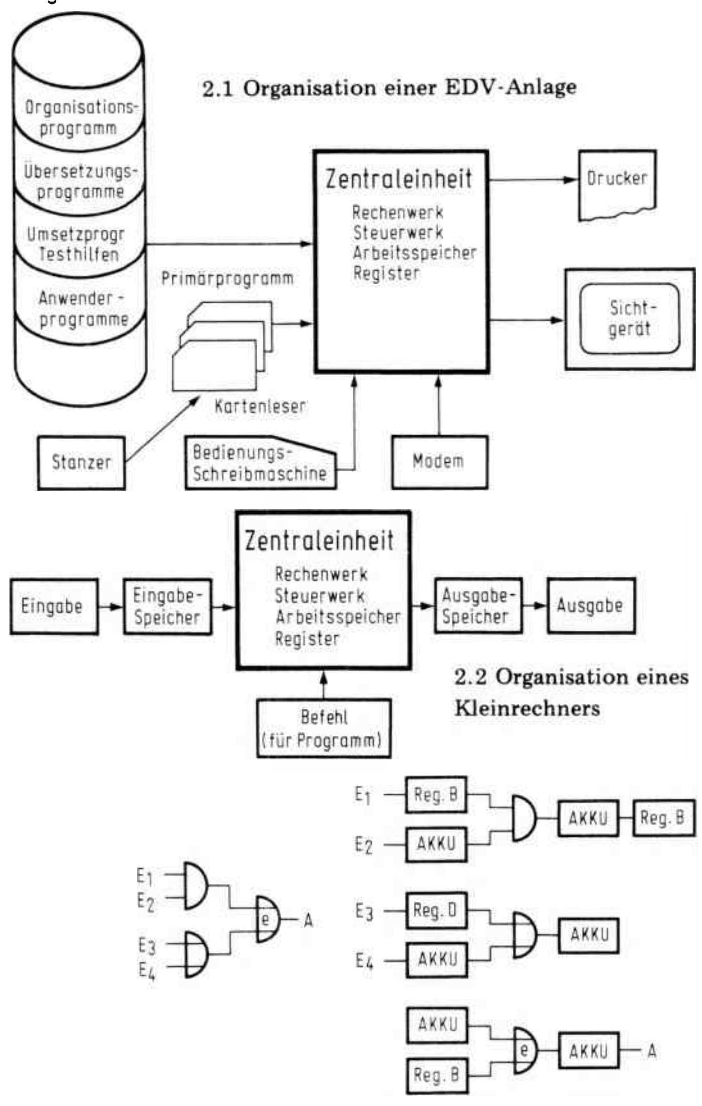
Die wichtigsten Register des Leitwerkes sind Befehlszähler, Befehlsregister, Indexregister, ggf. Indexarithmetik, Adressenregister und Operationsregister. Obwohl Leitwerke in ihrem Aufbau sehr viele Varianten aufweisen können, arbeiten sie prinzipiell wie folgt:

Die aus dem Arbeitsspeicher gelesenen Befehle werden im Befehlsregister bereitgestellt und jeweils in Adressteil, Operationsteil und Zusatzteil aufgespalten. Der Operationsteil wird in das Operationsregister übernommen und steuert von dort die Operationensteuerung an. Abhängig vom Zusatzteil wird bei Indizierung der Adressteil ggf. im Addierwerk des Leitwerkes zum Inhalt des angegebenen Indexregisters addiert oder es werden geforderte Substitutionen ausgeführt. Das Ergebnis dieser Adressenrechnung wird wieder in das Adressregister eingelesen, das den gewünschten Speicherplatz auswählt. Sobald der Befehl ausgeführt ist, wird der Inhalt des Befehlszählers um eins erhöht und geht dann in das Adressregister, wodurch das Lesen des nächsten Befehles eingeleitet wird [86].

Nun gibt es aber viele Anwendungsfälle, in denen keine so großen Anlagen benötigt werden, sondern ein Kleinrechner als Teil eines Prüfautomaten oder als Prozeßrechner in einer Fertigungsstraße eingesetzt wird. Eine solche Einheit sieht im allgemeinen folgendermaßen aus (*Abb. 2.2*):

Die Eingabe geschieht über den Eingabespeicher. Die Zentraleinheit enthält wieder das Rechenwerk, das Steuerwerk, den Arbeitsspeicher und Register. Der Befehl für ein bestimmtes Programm muß der Zentraleinheit zugeführt werden. Die Ausgabe erfolgt über den Ausgabespeicher.

Bei Datenverarbeitungsanlagen der zweiten und dritten Generation wurde der Prozessor, bestehend aus Rechenwerk und Leitwerk, speziell für die Anforderung der betreffenden EDV-Anlage entwickelt und verdrahtet. In Anlagen der zweiten Generation wurden Transistoren eingesetzt und in Anlagen der dritten Generation integrierte Schaltkreise, noch meistens im Bereich "small scale bzw. medium scale integration".



2.1 Organisation einer EDV-Anlage

2.2 Organisation eines Kleinrechners

2.3 Beispiel einer logischen Verknüpfung, links mit fest verdrahteter Logik, rechts Ablauf bei Mikrocomputern

Da einerseits mit dem Übergang zur LSI (large scale integration) die betreffenden Funktionen in einem einzigen Halbleiterkristall untergebracht werden konnten, andererseits deren Anwendungsbereiche sich von der Datenverarbeitungsanlage bis zum Taschenrechner "herunterschrumpfte", gelangte man zu einem Punkt, bei dem für jeden speziellen Anwendungsfall eine kundenspezifische Halbleiterschaltung erforderlich war.

Bekanntlich braucht die Erstellung einer kundenspezifischen Integration einen gewissen Zeitaufwand, der trotz weitgehender Automatisation durch CAD nicht mehr zu unterschreiten ist. Es ist auch unrationell, für jede bestehende Aufgabe eine spezifische integrierte Schaltung zu entwickeln.

Daher ist man seit etwa zwei Jahren einen anderen Weg gegangen:

Im Mikroprozessor steht ein kleines Rechen- und Leitwerk zur Verfügung, dessen interner Ablauf von außen mit einem programmierbaren Festwertspeicher beeinflußbar ist. Nun kann sich der Anwender durch geschicktes Programmieren des Festwertspeichers für jede spezielle Aufgabe sein passendes Logiknetzwerk selbst zusammenstellen.

Als Beispiel soll die boolesche Verknüpfung von vier Eingangsvariablen nach Abb. 2.3 (links) dienen. Die Variablen E_1 und E_2 sind durch ein UND-Glied, die Variablen E_3 und E_4 durch ein ODER-Glied verknüpft. Das Ergebnis dieser beiden Verknüpfungen geht auf ein Exklusiv-ODER-Gatter.

Eine fest verdrahtete Logikschaltung führt diese Verknüpfung parallel durch; ein Mikrocomputer muß die Operationen seriell abarbeiten, wobei die Zwischenwerte jeweils gespeichert werden. Das sieht dann, vereinfacht dargestellt, folgendermaßen aus:

Abb. 2.3, rechts (erste von oben)

0	IN	1;	Eingang 1(E_1) lesen, Daten stehen im Akkumulator
1	MOV	B,A;	Daten von Akkumulator in Register B schieben
2	IN	2;	Der Akkumulator wird mit den Daten des Eingangs E_2, der die Adresse 2 bekommt, geladen
3	ANA	B;	UND-Verknüpfung des Inhaltes von Register B mit dem Inhalt des Akkumulators. Das Ergebnis steht im Akkumulator

	4	MOV	B,A;	Das Ergebnis wird in das Register B geschoben und dort gespeichert. Es könnte ebenso dafür ein Speicherplatz im RAM genommen werden.
	5	IN	3;	Eingang 3 (E_3) lesen, Daten stehen im Akkumulator
	6	MOV	D,A;	Inhalt des Akkumulators in Register D sieben
Abb. 2.3, rechts (Mitte)	7	IN	4;	Der Akkumulator wird mit den Daten des Eingangs 4 (E_4) geladen
	8	ORA	D;	ODER-Verknüpfung des Inhalts von Register D mit dem Inhalt des Akkumulators. Das Ergebnis steht im Akkumulator.
Abb. 2.3, rechts (unten)	9	XRA	B;	Exklusiv-ODER-Verknüpfung des Inhalts vom Akkumulator (Ergebnis der zweiten Verknüpfung) mit dem Inhalt des Registers B, (Ergebnis der ersten Verknüpfung) Das Ergebnis steht im Akkumulator
	10	OUT	5;	Der Inhalt des Akkumulators wird auf den Ausgang gegeben.
	11	JMP	0;	Die Verknüpfungen werden zyklisch wiederholt, da sich die Eingangsvariablen ändern können.

Man sieht, daß die serielle Abarbeitung dieser einfachen Verknüpfung 12 Schritte erfordert. Die ersten Ziffern sind jeweils die Adressen für das ROM oder PROM, in dem die Befehle binär gespeichert sind. Im Falle des Mikroprozessors 8080, für den hier als Beispiel die mnemotechnischen Abkürzungen gewählt wurden, können die Adressen für die Ein- und Ausgänge die gleichen Nummern wie die für den ROM-Programmspeicher haben.

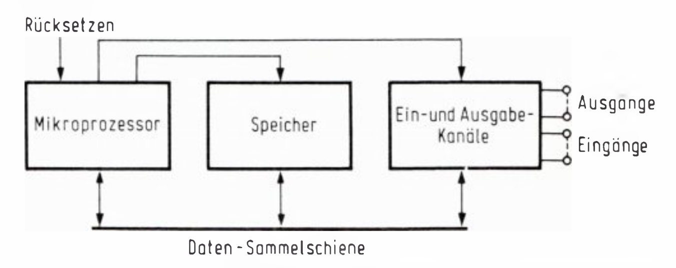
In der Praxis ist es natürlich für ein solch einfaches Beispiel nicht wirtschaftlich, einen Mikrocomputer einzusetzen. Man wird kleinere Logikschaltungen nach wie vor mit Standard-Logik-Bausteinen aufbauen. Aber der Mikrocomputer kann

nicht nur boolesche Verknüpfungen, sondern auch Additionen, Subtraktionen und viele andere Operationen ausführen. Die wirtschaftliche Grenze zum Übergang auf den Mikrocomputer liegt heute je nach den auszuführenden Operationen bei 30-90 MSI-Logik-Bausteinen.

3 Aufteilung in einem Mikrocomputer-System

Abb. 3.0.1 zeigt die vereinfachte Darstellung eines Computersystems mit einem Mikroprozessor. Als Speicher wird mindestens ein Schreib-Lesespeicher (RAM) benötigt. In diesem Falle muß das Programm nach jeder Versorgungsspannungsabschaltung neu eingelesen werden. In der Regel benutzt man zusätzlich zum RAM für das Programm einen Festwertspeicher (ROM). Steht das endgültige Programm noch nicht fest, so kann man zunächst ein RAM einsetzen, um nach Erprobung des Programmes sich von einem Halbleiterhersteller das passende ROM anfertigen zu lassen. Ferner steht in Form der programmierbaren Festwertspeicher (PROM) ein Bauelement zur Verfügung, in das sich der Anwender das Programm selbst einschreiben kann.

Bei den PROM's gibt es zwei Ausführungen. Bei der einen wird durch einen Stromimpuls innen eine Leiterbahn durchgeschmolzen oder eine Diode zerstört und damit der Speicher programmiert. Er läßt sich nachher nicht mehr umprogrammieren. Bei der zweiten Ausführung bekommt ein offenes "Gate" (floating gate) eine Ladung aufgeprägt, die sich nach Angabe der Halbleiterhersteller mehrere Jahrzehnte hält. Die Information kann durch Bestrahlung mit ultraviolettem Licht wieder gelöscht werden und in den Speicher ein neues Programm eingeschrieben werden.



3.0.1 Vereinfachte Darstellung eines Mikrocomputersystems mit einem Mikroprozessor

Außer dem Mikroprozessor und den Speichern wird zum Bau eines Mikrocomputers eine "Interface"-Logik benötigt. Sie dient dem Datenfluß von und zu den Ein- und Ausgabeeinheiten. Diese peripheren Bausteine müssen an den Prozessortyp angepaßt sein.

Der Mikroprozessor kann leider nicht so einfach wie eine gewöhnliche IS mit einem Datenblatt beschrieben werden. Ein internes Mikroprogramm oder ein fester Befehlsvorrat steuert nämlich das recht komplexe Gebilde dieses Mikroprozessors. Dabei müssen sowohl die von außen kommenden Signale wie "RESET", Unterbrechungsanfrage "INT", direkter Speicherzugriff nach "HOLD" und Speichersynchronisation über "READY" als auch die aus dem Prozessor kommenden Steuer- und Rückmeldesignale berücksichtigt werden. Dem Baustein steht nur eine begrenzte, durch das Gehäuse gegebene Anzahl von Anschlüssen zur Verfügung. Daher wird die Datensammelschiene zur Übertragung von Steuer- und Rückmeldesignalen mitbenutzt (Zeitmultiplexverfahren). Ein getrennter Baustein dekodiert die über die Datensammelschiene kommenden Signale (siehe Abschn. 3.2), ehe sie zur Steuerung der Ein- und Ausgabekanäle und Speicherbausteine benutzt werden. Die exakte Ausführung aller Befehle und Hilfsfunktionen des MP hängt von der richtigen "Software", d.h. von der Erstellung des dazugehörigen Programmes ab.

An Hand von drei Beispielen wird nachfolgend die Ergänzung des Mikroprozessors zu einem komplett funktionsfähigen System besprochen.

Grundsätzlich arbeiten alle Computer, so auch die Mikrocomputer, mit einem nach v. Neumann benannten Zyklus, der folgende Reihenfolge einhält:

1. Befehl holen
2. Befehl dekodieren
3. Operand(en) holen
4. Befehl ausführen und Ergebnis speichern oder ausgeben
5. Befehlszähler erhöhen

Signalfluß innerhalb des Mikrocomputers

Während eines Programmes treten zwischen den Mikrocomputer-Bausteinen folgende Betriebszustände und Signalflüsse auf:

1a. Befehlsabruf (Abb. 3.0.2). Der Mikroprozessor gibt über den Adreßbus die Adresse der Speicherstelle an das (P) ROM. Das (P) ROM liefert den codierten Befehl über den Datenbus an den Mikroprozessor.

1b. Festdatenabruf (Abb. 3.0.2). Der Mikroprozessor gibt über den Adreßbus die Adresse der Speicherstelle an das (P) ROM. Das (P) ROM liefert die Daten über den Datenbus an den Mikroprozessor.

2. Eingaben (Abb. 3.0.3). Der Mikroprozessor gibt über den Adreßbus die Adresse des Eingangskanals an die Eingabe-Einheit. Die Eingabe-Einheit transferiert die an den Eingangskanälen stehenden Daten an den Datenbus. Über den Datenbus gelangen die Daten in den Mikroprozessor.

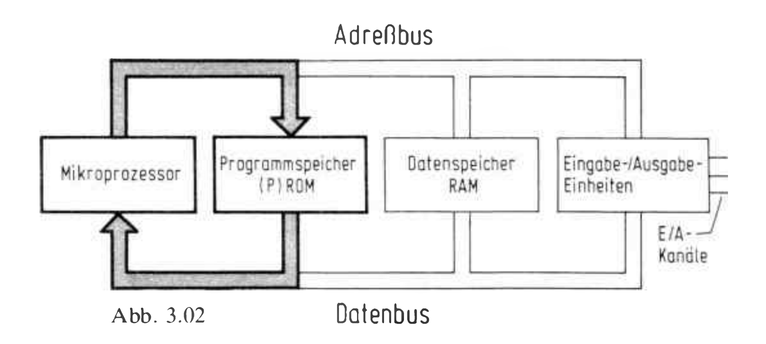


Abb. 3.02

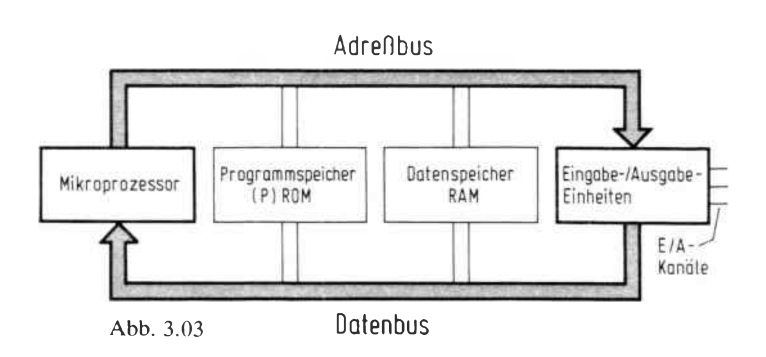


Abb. 3.03

3. Ausgaben (Abb. 3.0.4). Der Mikroprozessor liefert Adressen und Daten. Die Adressen gehen über den Adreßbus, die Daten über den Datenbus zur Ausgabe-Einheit.
   Ein- und Ausgabe-Einheiten können entweder in einem Baustein (SAB 8255) kombiniert oder mit getrennten Bausteinen (SAB 8212) ausgeführt sein.
4. RAM wird gelesen (Abb. 3.0.5). Der Mikroprozessor gibt über den Adreßbus die Adresse der Speicherstelle an das RAM. Das RAM liefert die Daten über den Datenbus an den Mikroprozessor.
5. Daten werden in das RAM geschrieben (Abb. 3.0.6). Der Mikroprozessor liefert Adressen und Daten. Die Adressen gehen über den Adreßbus, die Daten über den Datenbus in das RAM.

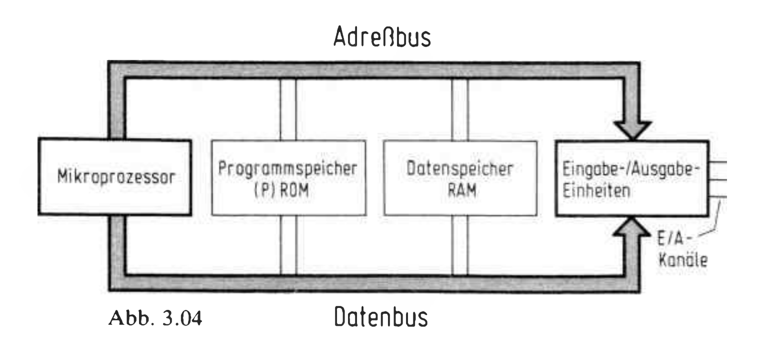


Abb. 3.04

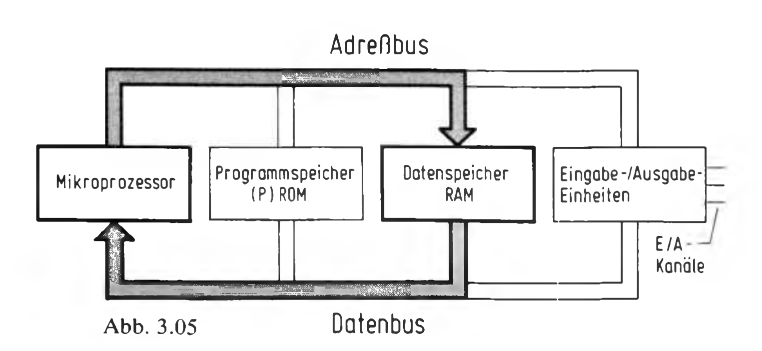


Abb. 3.05

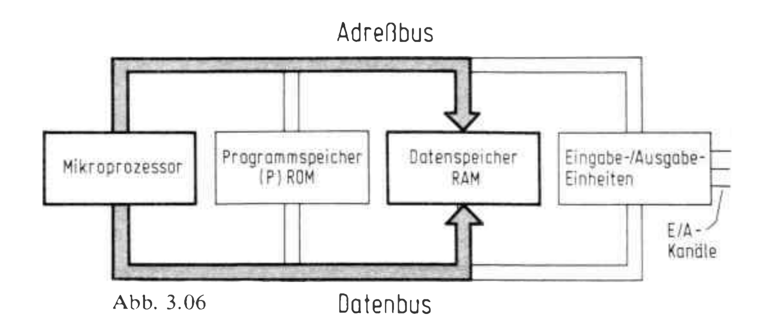
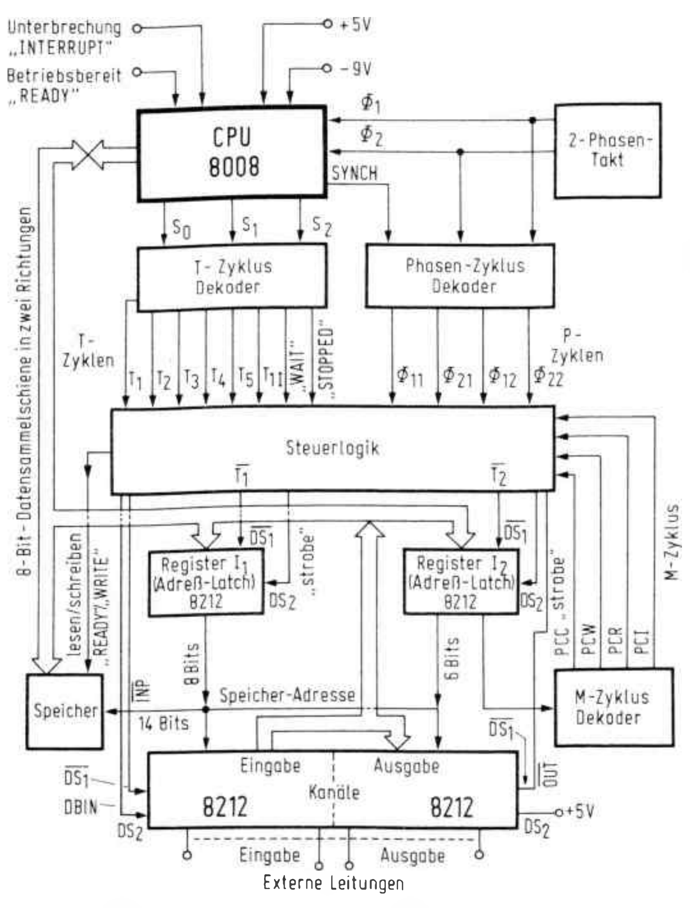


Abb. 3.06

Alle diese Vorgänge werden vom Mikroprozessor gesteuert. Nach dem RESET beginnt der Befehlszähler im allgemeinen bei der Adresse Null. Der Befehl wird aus dem ROM oder PROM gelesen. Die Dekodierung erfolgt im Mikroprozessor selbst. Da es sich bei Mikroprozessoren um "AKKU"-Maschinen handelt, muß immer ein Operand im AKKU stehen. Nach Ausführung des Befehls steht das Ergebnis im Akkumulator. Es kann in einem Register im Mikroprozessor gespeichert werden. Dies tut man, wenn man es für eine der nächsten Operationen gleich wieder braucht. Oder das Ergebnis wird in das RAM eingeschrieben, bzw. (bei der letzten Operation) ausgegeben.

3.1 Mikrocomputer mit der CPU 8008

Die CPU wird im Abschnitt 4.1 beschrieben. Zum besseren Verständnis wird aber mit der Erklärung des kompletten Systems begonnen. *Abb. 3.1.1* zeigt die Zusammenschaltung mit den übrigen Bausteinen. Eine 8-Bit-Datensammelschiene verbindet den Mikroprozessor mit den äußeren Speichern und den beiden Befehlsregistern I_1 und I_2. Diese Datensammelschiene kann Signale in beiden Richtungen übertragen. Es können sowohl Befehle und Daten aus dem Mikroprozessor in die Speicher und Register eingelesen, als auch umgekehrt aus diesen entnommen werden. Der Zeitmultiplexbetrieb gestattet die gleichzeitige Übertragung der Steuerinformationen, 14-Bit-Adressen und Daten zwischen der Zentraleinheit (CPU) und den externen Speichern auf einer Sammelschiene.



3.1.1 Ausgeführtes Mikrocomputersystem mit dem Mikroprozessor 8008, externe Beschaltung des Mikroprozessors

Tabelle 3.1.1

M-Zyklus	T-Zyklus	Operation
PCI	T_1	Letzte Stellen des Programmzählers nach I_1. Programmzähler zählt um einen Schritt weiter.
	T_2	Erste Stellen des Programmzählers und die Steuer-Bits nach I_2.
	T_3	Abruf der Befehle von den Speicherplätzen mit der Adresse aus I_1 und I_2. Einschreiben in das Befehlsregister des Mikroprozessors.
		Auslassen der T-Zyklen T_4 und T_5 und Sprung zum nächsten M-Zyklus.
PCR	T_1	Letzte Stellen des Programmzählers nach I_1. Programmzähler zählt um einen Schritt weiter.
	T_2	Erste Stellen des Programmzählers und die Steuer-Bits nach I_2.
	T_3	Lesen der Daten aus den Speicherplätzen mit der Adresse aus I_1 und I_2. Einschreiben in das interne Hilfsregister "b".
		Auslassen der T-Zyklen T_4 und T_5 und Sprung zum nächsten M-Zyklus.
PCW	T_1	Internes Register L wird nach I_1 ausgegeben (letzte Stellen der Adresse).
	T_2	Internes Register H und Steuerbits werden nach I_2 ausgegeben (erste Stellen der Adresse).
	T_3	Inhalt des internen Registers "b" geht zu den Speicherplätzen mit den Adressen aus I_1 und I_2.
		Auslassen der T-Zyklen T_4 und T_5 und Sprung zum nächsten M-Zyklus
Ende der Befehlsausführung		

Maschinen-Zyklen

Der v. Neumann-Zyklus enthält Teilbefehle, die beim Mikroprozessor 8008 mit den folgenden Maschinen-Zyklen (M-Zyklen) abgedeckt werden:

PCI	≙	Befehlsabruf
PCR	≙	Speicher lesen
PCW	≙	Speicher einschreiben
PCC	≙	Ein-Ausgabe

Jeder M-Zyklus setzt sich aus wiederum drei bis fünf T-Zyklen (im Englischen "states") zusammen. (In Tabelle 3.1.1 finden Sie ein Beispiel mit dem Einlesebefehl LMI, das später noch ausführlicher besprochen wird).

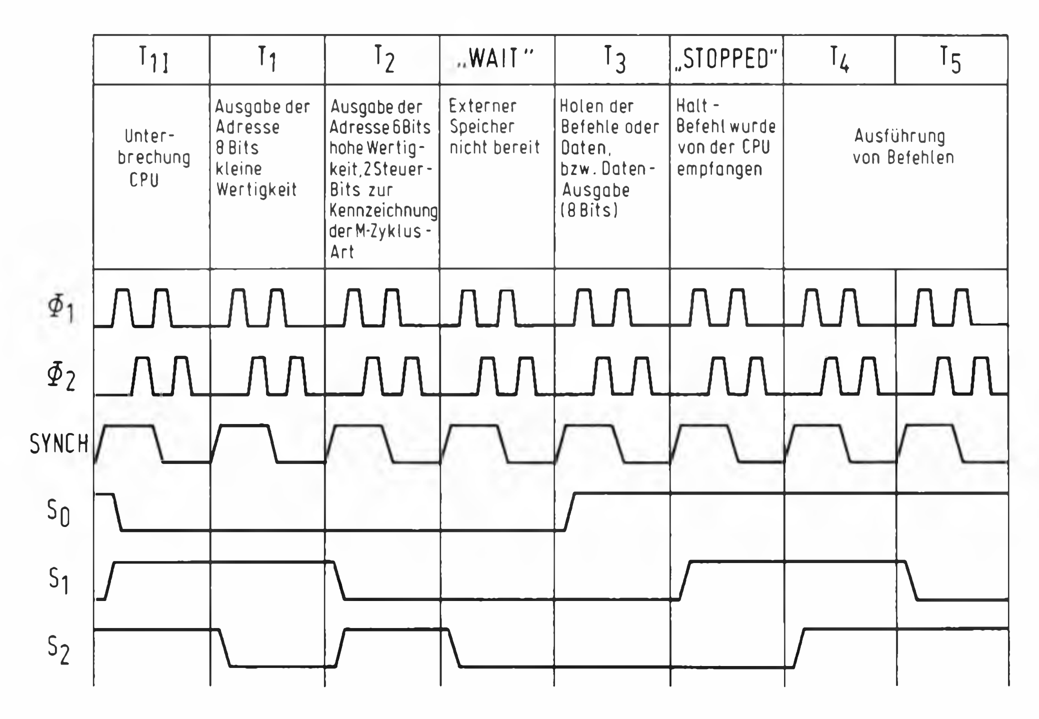
Damit die an den Mikroprozessor angeschlossenen Bausteine wissen, wer angesprochen wird, muß der Mikroprozessor Signale aussenden, welcher M-Zyklus folgt. Das geschieht über die Datensammelschiene über die Leitungen D_6 und D_7 während des T-Zyklus T 2. Über das Register I_2, welches als Auffangspeicher arbeitet, gelangen sie auf den M-Zyklus-Dekoder, der ein Steuersignal an die Steuerlogik abgibt und damit sagt, um welchen M-Zyklus es sich handelt (Tabelle 3.1.2).

Tabelle 3.1.2

M-Zyklus	Steuer-Bits MSB	Steuer-Bits LSB	
PCI	L	L	Befehlsabruf
PCR	H	L	Speicher lesen
PCW	H	H	Speicher einschreiben
PCC	L	H	Ein-Ausgabe

T-Zyklen

Man unterscheidet acht verschiedene T-Zyklen *(Abb. 3.1.2)*, von denen normalerweise drei bis fünf für einen M-Zyklus gebraucht werden. Die verbleibenden Zyklen können für "Unterbrechung" (T_{1I}), direkten Speicherzugriff (WAIT) und für den Prozessorhalt (STOPPED) benutzt werden. Bei der üblichen Operation werden während T 1 acht Bits von der CPU auf die Datensammelschiene gegeben. Da die Speicheradresse 14 Bits



3.1.2 Impulsplan für die verschiedenen T-Zyklen

umfaßt, werden zwei T-Zyklen für die Übertragung der kompletten Speicheradresse benötigt. Während T 1 werden die letzten Stellen, während T 2 die ersten Stellen und zwei Steuerbits für die Art des M-Zyklus übermittelt.

Beide Informationen werden in den Registern I_1 und I_2 gespeichert, bis durch Dekodierung der beiden Steuer-Bits (siehe Tabelle 3.1.2) bekannt ist, um welchen M-Zyklus es sich handelt. Im Zyklus T 3 holt sich der Mikroprozessor entweder ein Datenbyte, ein Befehlsbyte, oder gibt ein Datenbyte aus. Diese Übermittlung wird wieder über die Datensammelschiene zweigleisig je nach dem auszuführenden Befehl vorgenommen.

Während T 4 und T 5 führt der Mikroprozessor nun den Befehl aus und transferiert Daten zwischen den internen Registern. Einige Befehle brauchen diesen Zyklus nicht. In diesen Fällen werden die Zyklen übersprungen.

Tabelle 3.1.3 zeigt die Kodierung der einzelnen T-Zyklen mittels der Status-Bits S_0 bis S_2. Nach Dekodierung im T-Zyklus weiß die Steuerlogik genau, an welcher Stelle des Zeitablaufes sich der Prozessor befindet.

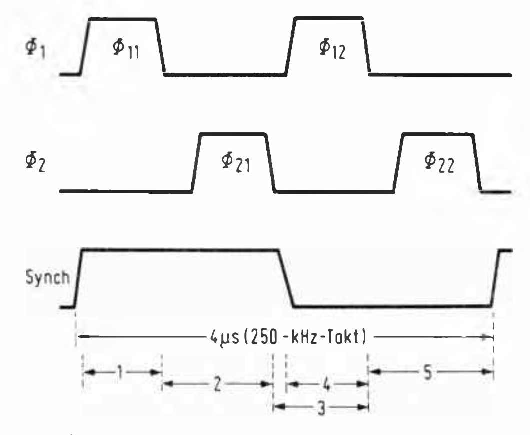
Zeitablauf

Das Flußdiagramm in *Abb. 3.1.3* zeigt einen M-Zyklus mit den möglichen Entscheidungen. Nach T 3 und T 4 wird geprüft, ob der vorliegende Befehl bereits ausgeführt wurde, was von der Art des M-Zyklus abhängt. Liegt kein Unterbrechungsbefehl "INTERRUPT" vor, beginnt der nächste M-Zyklus wieder mit T 1. Werden T 4 und T 5 übersprungen, arbeitet der 8008 asynchron in bezug auf die Zykluslänge.

Tabelle 3.1.3

T-Zyklus	Status Bits		
	S_0	S_1	S_2
T_1	L	H	L
T_2	L	L	H
T_3	H	L	L
T_4	H	H	H
T_5	H	L	H
T_{1I}	L	H	H
"WAIT"	L	L	L
"STOPPED"	H	H	L

3.1.3 Flußdiagramm der einzelnen T-Zyklen beim 8008



3.1.4 Ablauf der einzelnen Phasen während eines Taktes

"READY"-Befehlsleitung

Der 8008 kann mit jedem handelsüblichen Halbleiterspeicher betrieben werden. Schnelle Speicher haben die Daten immer sofort nach dem Senden des zweiten Adreßbytes bereit. In diesem Falle wird der Anschluß 17 (READY) mit U_{DD} verbunden und nicht benötigt. Bei langsamen Speichern sind die Daten nicht sofort verfügbar. Der Speicher gibt das Signal "$\overline{\text{READY}}$" ab, der 8008 muß warten, bis die angesprochenen Daten aus dem Speicher verfügbar sind. Die CPU verbleibt so lange in dem Zustand "Warten" (WAIT), bis der Speicher mittels des "READY"-Signales die Bereitstellung der Daten anzeigt. Er geht dann in den T-Zyklus T 3 über.

"INTERRUPT"-Befehlsleitung

Die Steuerleitung "Unterbrechung" (INTERRUPT) darf zu jeder Zeit eines Befehles aktiviert werden. Der Mikroprozessor führt zunächst noch den verbleibenden Zyklus des Befehles aus, geht dann aber in den sogenannten T_{1I}-Zyklus über. Damit kann man von außen einen RST-Befehl während des Zyklus T3 einfügen. Dieser Befehl ruft einen der acht Speicherplätze auf, in dem das Unterprogramm zur Unterbrechung steht. Bei einer Unterbrechung zählt der Programmzähler nicht weiter, sondern sein Inhalt wird in einen internen Stapel geschoben. Daher sollte ein "RETURN"-Befehl immer am Ende einer Unterbrechung programmiert sein. Diese Programmierart bewirkt, daß der Inhalt des Programmzählers automatisch nach der Ausführung wieder in den Originalzustand überschrieben wird und die normale Operation fortgeführt wird.

Taktsteuerung

Die gesamte Funktion der Prozessors steht und fällt mit der Taktsteuerung.

Die Taktphasen sind $\emptyset_1$ und $\emptyset_2$ (siehe Abb. 3.1.2). In der CPU wird die Phase 2 intern durch zwei geteilt und daraus der sogenannte "SYNCH"-Impuls abgeleitet. der SYNCH-Impuls ist für die Außenbeschaltung verfügbar. Eine volle Periode des Synchimpulses wird T-Zyklus genannt. Dieser setzt sich wiederum aus vier sequentiellen Impulsen *(siehe Abb. 3.1.4)*, den Phasen $\emptyset_{11}$, $\emptyset_{21}$, $\emptyset_{12}$, $\emptyset_{22}$ zusammen. Jedem einzelnen Zeitabschnitt fällt folgende Aufgabe zu:

Abschnitt 1: Vorbereitung der internen Speicher und der Datensammelschiene
Abschnitt 2: Lesen des internen Speichers und Vorbereitung der Addierer
Abschnitt 3: Ausführung von arithmetischen und logischen Operationen
Abschnitt 4: Vorbereitung des Einschreibens in den internen Speicher und der Datensammelschiene
Abschnitt 5: Schreiben und Wiederauffrischung des internen Speicherinhaltes.

Adreßstapel

Der in der CPU enthaltene Adreßstapel besteht aus acht 14-Bit-Registern. Ein Register wird als Programmzähler verwendet (Speicherung der effektiven Adresse) und die anderen sieben erlauben die Abstellung von Adressen während der Unterprogramme. Im Stapel wird automatisch der Inhalt des Programmzählers nach der Ausführung eines "RETURN's" wieder in den Programmzähler überschrieben. Die Register des Stapels werden als "zuletzt ein/zuerst heraus"-Rückstellspeicher verwendet. Eine 3 Bit-Adressenanzeige gibt an, welches der acht Register gerade der Programmzähler ist. Wenn die Kapazität des Stapels überschritten wird, stellt sich die Adressenanzeige zurück und der Inhalt des "Lowest-Level"-Registers, d.h. des Registers, bei dem die Adresse zuerst eingelesen wurde, wird zerstört. Der Programmzähler wird sofort nach der Aussendung der Adreßbits niedriger Ordnung erhöht. Die Adreßbits hoher Ordnung werden während T 2 gesendet und dann erhöht, wenn ein Übertrag aus T 1 resultiert. Der 14-Bit-Programmzähler erlaubt die direkte Adressierung von 16 k-Bytes im Speicher. Durch den Gebrauch von Eingabe/Ausgabe-Befehlen zur Umschaltung kann die Speicherkapazität erweitert werden.

Starten des 8008

Wenn die Stromversorgung (U_{DD}) und die Taktimpulse (Φ_1, Φ_2) zum ersten Mal angeschaltet werden, so wird ein internes Flip-Flop beim Anstieg von U_{DD} gesetzt. Dieses interne Signal bringt ein "HALT" (LLLLLLLL) in das Befehlsregister und der 8008 ist dann im "STOPPED"-Zustand. Die folgenden 16 Takt-Perioden nach Eintreten in den "STOPPED"-Zustand wer-

den benötigt, um die Speicher auf logisch "L" zu setzen (Akkumulator, Scratch Pad, Programmzähler und Stapel). Während dieser Zeit war die "Interrupt"-Leitung auf logisch "L". Zu jeder beliebigen Zeit nachdem die Speicher auf logisch "L" gesetzt sind, ist der 8008 zur normalen Arbeit bereit.

Um das Flip-Flop rückzusetzen und auch wieder aus dem "STOPPED"-Zustand herauszubekommen, muß die "INTERRUPT"-Leitung auf logisch "H" gelegt werden. Sie sollte wieder auf logisch "L" gesetzt werden, wenn der Zyklus T_{1I} dekodiert wird. Man beachte, daß immer im T_{1I}-Zyklus der Programmzähler nicht fortgeschaltet wird. Demzufolge wird die gleiche Adresse in zwei aufeinanderfolgenden Zyklen gesendet. Jetzt wird die Adresse des Speicherplatzes Null für das erste Befehls-Byte genommen. Nachdem der erste Befehl in den Platz Null geladen ist, wird das Befehls-Byte während des Zyklus T 3 abgerufen. Sodann führt der Mikroprozessor nacheinander alle programmierten Befehle und Abzweigungen aus.

Drei Abläufe sind möglich, um den 8008 zu starten. Der "RESTART"-Befehl ist ein Ein-Zyklus-Befehl und man sollte diesen Befehl nur benutzen, um ein Anfangs-Unterprogramm aufzurufen. Sonst ist es nicht notwendig, den 8008 mit einem "RESTART"-Befehl zu starten.

Die Auswahl der Starttechnik, die man verwendet, hängt von dem Aufbau des Systems um den 8008 ab. Wenn der "INTERRUPT" nur für den Start des 8008 verwendet werden soll, benutzt man am besten den ROM selbst. Wenn der "INTERRUPT" verwendet wird, um Befehle in den 8008 zu bringen, ist es sinnvoll, ihn auch zu benützen, um den Anfangsbefehl zu geben.

Beispiel: Einlesebefehl "LMI"

Tabelle 3.1.1 zeigt die Abarbeitung des Einlesebefehles "LMI" (lade den Speicher sofort mit den nächsten Byte, welches auf dieses Byte folgt). Der Einlesebefehl benötigt die Plätze von zwei Bytes im Speicher, ihre Ausführung braucht drei Zyklen. Bei Beginn des Befehles und während des Zyklus T 1 werden die acht letzten Bits des Programmzählers dem Register I_1 zugeführt. Der Programmzähler zählt um einen Schritt weiter. Die sechs Bits des Programmzählers mit der höchsten Wertigkeit und die beiden Steuer-Bits gelangen zum Register I_2 während

des Zyklus T 2. Am Ende von T 2 gibt der Dekoder ein PCI-Signal an die Steuerlogik, das anzeigt, daß es sich um eine Befehlsabrufperiode handelt. Während des Zyklus T 3 werden die Speicherplätze mit dem Inhalt der Register I_1 und I_2 adressiert und der Speicherinhalt in das Befehlsregister des Mikroprozessors eingeschrieben. Der Befehlsdekoder erkennt, daß es sich um einen direkten Befehl und eine Transfer-Steuerung für das eigene Mikroprogramm handelt. Wenn die Zyklen T 4 und T 5 nicht benötigt werden, endet der Umlauf mit T 3.

Ähnlich wird die zweite und dritte Umlaufoperation ausgeführt. Hilfsregister b bekommt im zweiten Umlauf seine Information. Hilfsregister a und b werden von dem Mikroprogramm gebraucht, um Daten intern zu transferieren. Im dritten Umlauf werden die Register H und L – es sind zwei der sieben 8-Bit-Register A bis L, die auf dem Prozessor-Chip untergebracht sind – angesprochen. Register A ist der Akkumulator, die anderen sind Pufferspeicher.

Unterprogramm zur Verbindung der Register H und L

Die Register H und L werden über ein Unterprogramm miteinander verbunden. Um einen Speicherreferenzbefehl ausführen zu können, nimmt die Logik die 14 Bit-Speicheradresse aus dem Inhalt des Registers H (sie enthält die 6 Bits mit der höchsten Wertigkeit) und aus dem Register L (die 8 Bits mit der kleinsten Wertigkeit). Die Befehle enthalten kein Feld für die Speicheradresse. Da die Register H und L sowohl inkrementiert und dekrementiert werden, als auch logisch und arithmetisch mit dem Akkumulator (Register A) zusammenarbeiten können, ist es möglich, sämtliche Speicherplätze abzufragen und zu indizieren. Wenn z.B. das Register L während dieses Vorganges durch Null geht, muß das Register H inkrementieren oder dekrementieren, um den Abfragevorgang fortsetzen zu können.

Wie schon oben erwähnt, sind im Mikroprozessor die beiden Register H und L nur über ein Unterprogramm miteinander verbunden. *Tabelle 3.1.4* zeigt solch ein Unterprogramm für ein Inkrement und Dekrement aller 16 Speicherplätze. Der Inhalt der Register H und L ist dabei von der internen Befehlsadresse und dem Programmzähler unabhängig.

Tabelle 3.1.4: Unterprogramm zur Verbindung der internen Register H und L

	Wortsymbol	Befehle	Kode	Kommentar
Inkrement der Register	INCHL	INL	L L H H L L L L	Inkrement Register L
		RFZ	L L L L H L H H	Rückkehr, wenn keine Null
		INH	L L H L H L L L	Inkrement Register H
		RET	L L X X X H H H	Rücksprung
Dekrement der Register	DECHL	DCL	L L H H L L L H	Dekrement Register L
		CPI	L L H H H H L L	Vergleiche mit dem nächsten Byte
		377	H H H H H H H H	Alles auf „eins"
		RFZ	L L L L H L H H	Rückkehr, wenn nicht gleich
		DCH	L L H L H L L H	Dekrement Register H
		RET	L L X X X H H H	Rücksprung

X = Beliebig

Ein- und Ausgabe

Der PCC-Zyklus wird für die Ein- und Ausgabe benötigt. Sie setzt sich aus einem PCI-Zyklus, bestehend aus drei T-Zyklen für den Befehlsabruf, gefolgt von einem PCC-Zyklus, zusammen. Am Ende von T 1 hat das externe Register I_1 den Inhalt des Registers A (Akkumulator). Das externe Register I_2 hat die 6 Bits des Befehlsbyte mit der kleinsten Wertigkeit, welches während T 2 übergeführt wurde. Das Befehlsbyte enthält ein Bitfeld, welches die entsprechenden Ein- und Ausgabeeinheiten adressiert. Im Befehl ist ferner zu erkennen, ob es sich um eine Ein- oder Ausgabeoperation handelt. Wenn eine Ausgabeoperation aufgerufen wird, wird T 3 nicht gebraucht. Jetzt muß der Inhalt des Registers I_1 (vorher im Akkumulator) zu einem der 24 möglichen Ausgangseinheiten transferieren. Die Einheit wird durch das Zeigerfeld im Register I_2 bestimmt. Register I_1 enthält die Datenbytes, die vor dem Ausgabebefehl im Register A waren. Der M-Zyklus endet mit T 3.

Um eine Eingangsoperation aufzurufen, muß das Eingangsbyte auf der Datensammelschiene vor dem Zyklus T 3 vorhanden sein. Während T 4 werden "bedingte Kennzeichen" auf der Datensammelschiene sein. Diese werden für Testzwecke oder Anzeige des T-Zyklus hergenommen. Am Ende von T 5 sind die eingegebenen Datenbytes im Register A.

Die Eingangseinheit kann je nach Außenschaltung auf zwei Arten adressiert werden. Die eine nimmt das Zeigerfeld im Register I_2, ähnlich wie es während der Ausgangsoperation getan wird. Dabei kann nur eine der acht Eingangseinheiten adressiert werden. Die zweite Methode benutzt das vollständige 8 Bit-Byte im Register I_1 (anstatt des Zeigerfeldes im Register I_2), um eine der 256 möglichen Eingangseinheiten zu adressieren. Während der Inhalt des Registers A im Zyklus T 1 in dem externen Register untergebracht ist, verlangt diese Technik, daß die Adresse der Einheit im Register A vorhanden sein muß, bevor der Eingangsbefehl ausgeführt werden kann.

Viele der Mikrocomputer-Geräte mit 48 Befehlen haben Ergänzungen, wie z.B. Register oder "bedingte Kennzeichen". Damit kann man auf 230 individuelle Befehlsbytes erweitern.

3.2 Mikro-Computer mit Mikroprozessor 8080

Mit dem Mikroprozessor-Baustein 8080 steht heute eine universell einsetzbare CPU zur Verfügung. Gegenüber dem 8008, welcher in P-MOS-Technologie hergestellt wird, arbeitet der 8080 mit der schnelleren N-MOS-Technologie. Die Additionszeit im Register für ein Datenwort verkleinert sich damit von 20 auf 2 μs. Im Gegensatz zum 8008 besitzt der 8080 getrennte Daten- und Adressensammelschienen. Er hat eine Adressenkapazität von 64 kBytes.

Bausteine

Zum Aufbau eines Mikrocomputers werden zusätzlich zu der CPU die Speicher und Interface-Bausteine benötigt. Nachstehend die wichtigsten für das System 8080 verfügbaren Bausteine:

8101	RAM	256 x 4 Bit	
8111	RAM	256 x 4 Bit	
8102	RAM	1024 x 1 Bit	
8107A	RAM	4096 x 1 Bit	dynamisch
8302	PROM	256 x 8 Bit	maskenprogrammierbar
8308	ROM	1024 x 8 Bit	
8316	ROM	2048 x 8 Bit	
8702A	PROM	256 x 8 Bit	elektr. programmierbar, durch ultraviolette Strahlen löschbar
8604	PROM	512 x 8 Bit	schnelle Zugriffzeit 100 ns, elektr. programmierbar, nicht löschbar
8704	PROM	512 x 8 Bit	elektr. programmierbar, löschbar durch ultraviolettes Licht
8205	1 aus 8-Dekoder		
8212	steuerbarer Ein- und Ausgabespeicher		
8210	Pegelanpassung TTL zu MOS-Pegel		
8216	4-Bit-Sammelschienentreiber zweidirektional		
8201	Universal-Verbindungs-Interface		
8224	Takt-Generator		
8228	Systemsteuerung		
8255	programmierbare Daten-Interface		

Beispiele kompletter Mikro-Computer

Anhand von zwei Beispielen soll gezeigt werden, wie ein System, bestehend aus der CPU und den anderen Bausteinen, aufgebaut werden kann. Der Vergleich der beiden Abb. 3.2.1 und 3.2.11 zeigt, wie mit höher integrierten Bausteinen ein System immer einfacher und übersichtlicher wird.

Dateneingabe

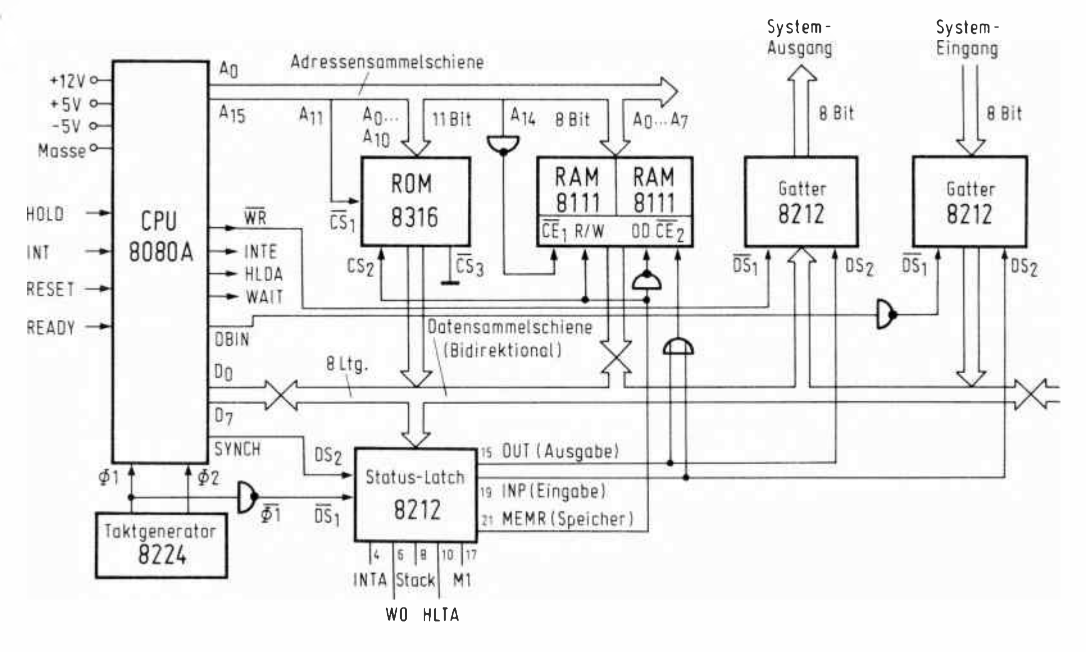
Im Beispiel *Abb. 3.2.1* werden die Daten über steuerbare Eingabespeicher 8212 eingegeben. Da jeder Speicher nur in der Lage ist, 8 Bit zwischenzuspeichern, müssen, wie später gezeigt, bei einer höheren Anzahl von Eingangsleitungen mehrere Bausteine eingesetzt und eine Auswahl getroffen werden. Die Aktivierung der Eingabespeicher erfolgt über die Steuerleitungen "INP"(Eingabe) vom Status-Latch und über DBIN (über einen Inverter) von der CPU.

Taktgenerator

Der Taktgenerator 8224 versorgt die CPU mit den beiden Taktphasen. Aus diesem Baustein steht ferner die direkte Oszillatorfrequenz, ein "STATUS"-Strobe-Impuls sowie ein in einem D-Flip-Flop gespeichertes "READY"-Signal zur Verfügung.

STATUS-Signale

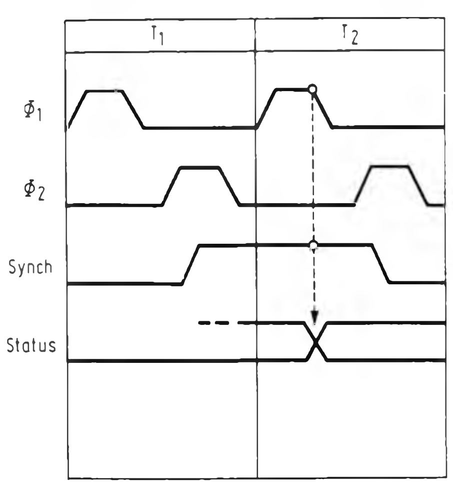
Der als "Latch" geschaltete Baustein 8212 dient zur Erweiterung der Ausgänge aus der CPU. Normalerweise müßte der 8080 mit einem 48-Pin-Gehäuse versehen sein. Da er nur über 40 Anschlüsse verfügt, hilft man sich so, daß über die Datensammelschiene Multiplex-Informationen ausgegeben werden, so daß die restlichen acht Informationen nach Dekodierung über den 8212 zwischengespeichert zur Verfügung stehen. *Abb. 3.2.2* zeigt die Signale auf der Datensammelschiene für die verschiedenen zu übertragenden Informationen. Aus der Bit-Kombination D_0-D_7 während der Überlappung der Phase Φ_1 und des Synch-Impulses *(Abb. 3.2.3)* im T 2-Zyklus ergibt sich damit auch der Typ des Maschinen-Zyklus. Folgende Status-Signale werden dekodiert:



3.2.1 Minimalkonfiguration für ein Mikrocomputersystem mit dem Mikroprozessor 8080 und Ein-Ausgabebausteinen 8212

Bits der Datensammelschiene	„Status"-Information	Befehl holen	Speicher lesen	Speicher einschreiben	Stapel lesen	Stapel schreiben	Eingang lesen	Ausgeben	Unterbrechungsquittung	Halt-Quittung	Unterbrechungsquittung während eines „Halt"
		M-Zyklus-Typ									
	T-Zyklus-Wort →	1	2	3	4	5	6	7	8	9	10
D_0	„INTA"	L	L	L	L	L	L	L	H	L	H
D_1	„$\overline{\text{WO}}$"	H	H	L	H	L	H	L	H	H	H
D_2	„STACK"	L	L	L	H	H	L	L	L	L	L
D_3	„HLTA"	L	L	L	L	L	L	L	L	H	H
D_4	„OUT"	L	L	L	L	L	L	H	L	L	L
D_5	M_1	H	L	L	L	L	L	L	H	L	H
D_6	„INP"	L	L	L	L	L	H	L	L	L	L
D_7	„MEMR"	H	H	L	H	L	L	L	L	H	L

3.2.2 Signale auf der Datensammelschiene zur Übertragung der fehlenden Status-Informationen



3.2.3 Zeitpunkt während T_2 zur Dekodierung der fehlenden Status-Informationen

INTA Quittung für Unterbrechungsgesuch.
Dieses Signal kann benutzt werden, um den Eingang eines Wiederanlaufbefehls in die Datensammelschiene zu verhindern, wenn "DBIN" aktiviert ist.

$\overline{WO}$ Negation von "write output" zeigt an, daß der laufende Maschinenzyklus Speicher-Einschreibefunktion hat oder die Ausgabe von Daten erfolgt.

STACK Stapel, zeigt an, daß die Adressensammelschiene die letzte Stapeladresse vom Stapelzeiger führt.

HLTA Quittung für HOLD-Befehl.

OUT Zeigt an, daß die Adressensammelschiene die Adresse der Ausgabe-Einheit führt und die Datensammelschiene die Ausgangsdaten haben wird, sobald $\overline{WR}$ aktiviert ist.

M 1 Liefert ein Signal, das anzeigt, daß die CPU sich die ersten Bytes eines Befehles holt.

INP Zeigt an, daß die Adressensammelschiene die Adresse einer Eingabeeinheit führt und die Daten sollten auf die Datensammelschiene gegeben werden, sobald "DBIN" aktiviert ist.

MEMR Zeigt an, daß die Datensammelschiene zum Lesen von Daten aus dem Speicher benutzt wird.

Mit der Kombination dieser acht STATUS-Signale und dem Signal "DBIN" können alle notwendigen Steuerungen zur Ein- und Ausgabe, Ansprechen der Speicher und Eingabe von "INTERRUPT"-Anweisungen ausgeführt werden.

Ausgabe

Sollen Daten ausgegeben werden, so erhalten die Ausgabespeicher 8212 das Ausgabesignal "OUT". Die Adresse wird über die Adressensammelschiene übertragen. Die Daten sind verfügbar, sobald $\overline{WR}$ aktiviert ist.

Speicher

In dem Festwertspeicher ROM 8316 ist das Programm gespeichert. Die Adressensammelschiene wählt den Speicherplatz aus. Die aktivierte Leitung "Speicher" (MEMR) sagt aus, daß ein Auslesen aus dem Speicher erfolgen soll. Die Leitung "DBIN" gibt ferner an, daß die Datensammelschiene zum Empfang von

Daten geöffnet ist. Speicherkapazitätserweiterungen sind durch die "Chip-Select-Eingänge" bei den hier angezogenen Halbleiterspeichern, wie später beschrieben, möglich.

Als Arbeitsspeicher dient das RAM. Hier erfolgt die Wahl, ob eingeschrieben oder ausgelesen werden soll, am R/W-Eingang. Eingeschrieben wird, wenn das Signal $\overline{MEMR}$ vorliegt. Jetzt kommen die Daten über die Datensammelschiene aus der CPU und werden in den Schreib-Lese-Speicher eingeschrieben. Daß der Speicher angesprochen ist, ergibt sich aus der richtigen Adresse der Adressensammelschiene. Um eine Verwechslung von Adressen für die Ein-Ausgabe und für die Arbeitsspeicher zu vermeiden, wird bei Ein- oder Ausgabe über das ODER-Gatter und den Chip-Select-Eingang $\overline{CE_2}$ das RAM gesperrt. Umgekehrt wird das Auslesen aus diesem Speicher durch das aus dem "Latch" kommende Signal "Speicher" (MEMR) signalisiert.

Unterbrechung für Unterprogramme

Die Eingabe von "INTERRUPT"-Befehlen in die Datensammelschiene wird durch die Leitung "INT" sinngemäß vorbereitet.

Datensammelschiene

Die Datensammelschiene kann für bestimmte Anwendungsfälle nach außen geführt werden. Dieses ist z.B. bei Mehrfachbetrieb notwendig. Hierbei ist besonders darauf zu achten, daß die Belastbarkeit der aus den Klemmen D_0 bis D_7 der CPU kommenden Leitungen genügend gesichert ist. Gegebenenfalls müssen zwischen der CPU und der Datensammelschiene entweder zwei Bausteine 8216 oder 8212 nach *Abb. 3.2.4* oder ein Baustein 8228 eingeschleift werden.

Maschinen-Zyklen

Es gibt 9 Arten von M-Zyklen (siehe auch Abb. 3.2.2). Jeder Befehl erfordert ein bis fünf Maschinenzyklen oder Speicherzyklen zum Holen und zur Ausführung. Befehle enthalten ein bis drei Bytes. Der M 1-Zyklus wird so oft wiederholt, wie es der Befehl erfordert. Jeweils nach dem letzten M-Zyklus wird geprüft, ob eine "Unterbrechungsanfrage" vorliegt.

T-Zyklen

Jede Taktperiode entspricht einem T-Zyklus. Jeder M-Zyklus erfordert drei bis fünf T-Zyklen, der M-Zyklus M 1 zum Holen der Befehle 4 bis 5, die Zyklen M 2 ... M 5 normalerweise drei T-Zyklen. Für sämtliche M-Zyklen werden minimal 4 T-Zyklen für einfache Befehle, wie arithmetische Operationen, die im Akkumulator ausgeführt werden können, benötigt, bis maximal 18 T-Zyklen für komplexe Befehle wie z.B. der Wechsel des Inhaltes der Register H und L.

Nur wenn die Zustände "Warten" (WAIT), "Belegt halten" (HOLD) oder "HALT" eingegeben werden, hängt die genötigte Zeit einzig und allein von den externen Geräten ab.

Zyklus T 1

Es wird der Inhalt des Programmzählers auf die Adressensammelschiene gegeben. Der SYNCH-Impuls geht auf "H". Die Datensammelschiene bekommt die "STATUS"-Information, die den beginnenden M-Zyklus betrifft.

Zyklus T 2

Während T 2 werden die "STATUS"-Informationen dekodiert und die Bedingungen "READY", "HOLD" und "HALT" abgefragt. Wenn die "READY"-Leitung H-Pegel führt, geht die CPU in T 3 über.

Zyklus T 3

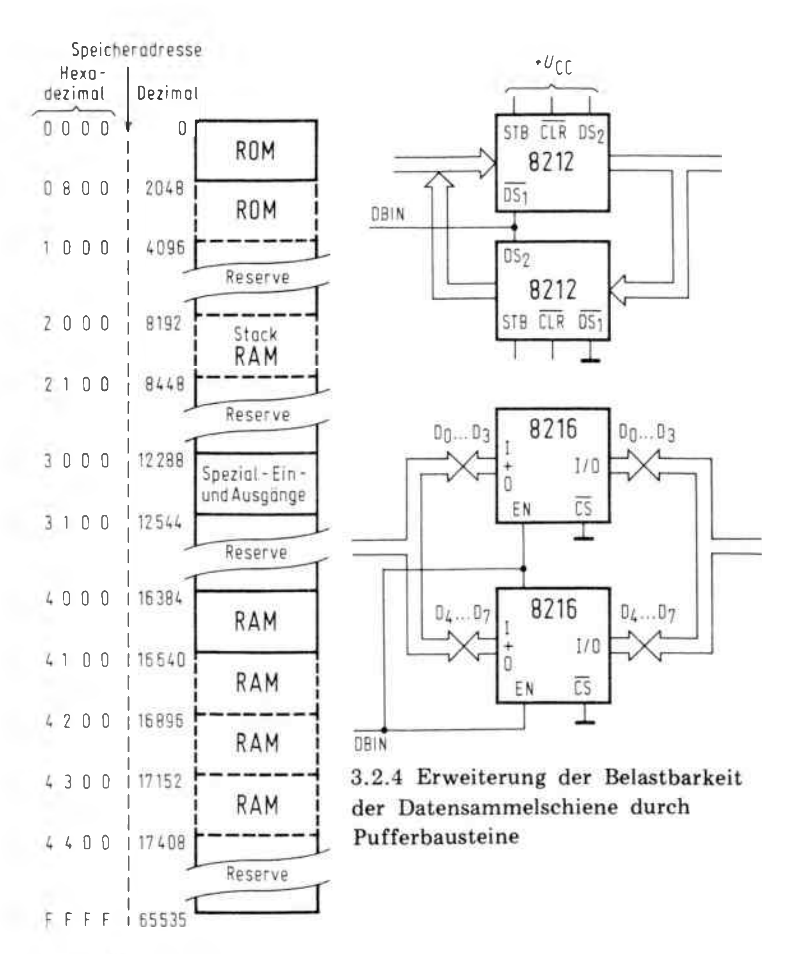
Holen der Daten oder Befehle bzw. Einschreiben oder Ausgeben der Daten.

Zyklus Tw

Führt die "READY"-Leitung im T 2 L-Pegel, gelangt die CPU in den Warte-Zyklus, bis die Daten abrufbereit sind und "READY" wieder H-Pegel führt. Dies dient zur Anpassung an langsame Speicher sowie langsame Ein- und Ausgabeeinheiten.

Zyklus T 4 und T 5

Beide Takte sind wahlweise vorgesehen, um — wenn nötig — die Befehle bis zum Ende auszuführen. Nach T 4 oder T 5 fängt der nächste M-Zyklus wieder mit T 1 an.



3.2.4 **Erweiterung der Belastbarkeit der Datensammelschiene durch Pufferbausteine**

3.2.6 **Vorschlag für die Speicheradressenbelegung bei einem ausbaufähigen Mikrocomputersystem**

Zeitablauf

Das Flußdiagramm in *Abb. 3.2.5* zeigt den Zeitablauf mit den möglichen Eingaben und Entscheidungen. Während T 2 wird über die "READY"-Leitung geprüft, ob die Daten aus dem Speicher oder Eingabeeinheiten verfügbar sind, bzw. der Speicher einschreibebereit oder die Ausgabeeinheiten ausgabebereit sind. Liegt keine "HALT"-Quittung vor, wird nach dem Belegt-Halte-Signal "HOLD" abgefragt. Dann geht der Lauf über T 3 (Bedarfsweise auch T 4 und T 5), über eine nochmalige interne Belegt-Abfrage zur Entscheidung, ob der Befehl ausgeführt ist, weiter. Liegt keine Unterbrechungsanfrage "INT" oder "INTE" vor, folgt der nächste T-Zyklus T 1.

"HLTA"-Entscheidung

Ist das "READY"-Signal nur wegen zu langsamer Speicher auf "L", so erfolgt keine "HALT"-Quittung "HLTA". Die CPU geht in den T_W-Zyklus, bis "READY" wieder H-Signal führt. Sonst geht sie in den T_{WH}-Zyklus, der sich bei negierten Signalen "$\overline{\text{HOLD}}$" oder "$\overline{\text{INT}}$" selbst aufrecht erhält. Bei belegt halten "HOLD" wird das "Belegt"-Flip-Flop gesetzt. Die CPU gerät in den Zustand "Belegt halten" (HOLD). Dieser Zustand wird durch löschen des Signales "HOLD" wieder aufgehoben.

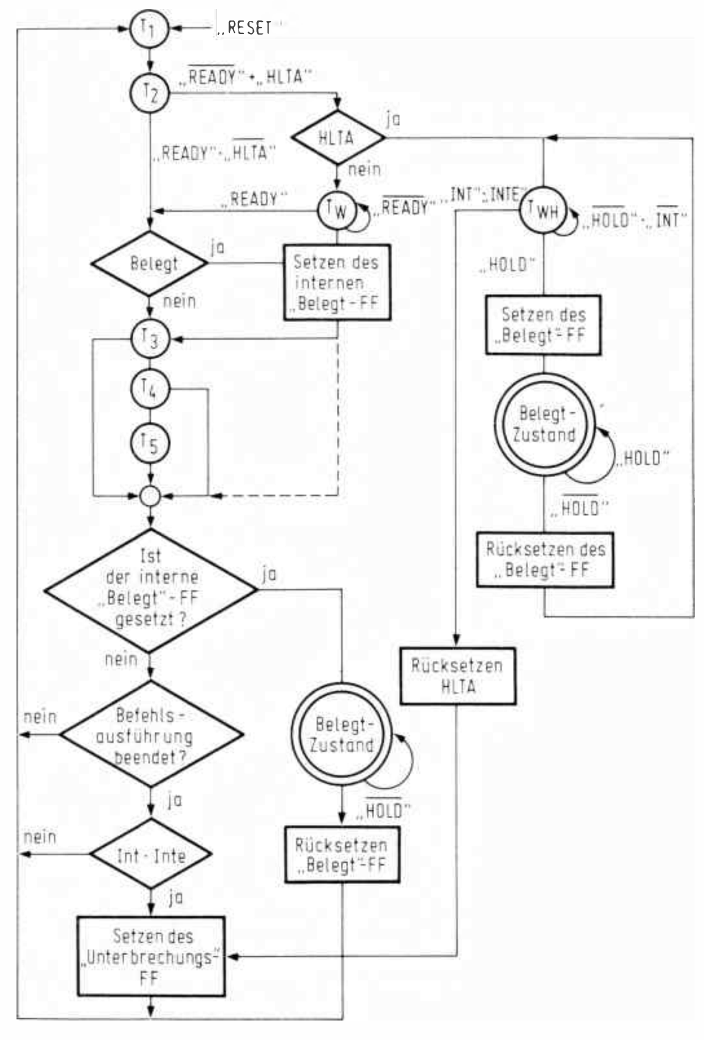
Sind aber die Signale "Unterbrechung" (INT oder INTE) vorhanden, wird die "HALT"-Quittung rückgesetzt und dafür das "Unterbrechungs"-Flip-Flop "INT-FF" gesetzt.

"RESET"

Ein Wiederanlauf läßt sich nur mit dem "RESET"-Signal erreichen.

"Belegt-Halten"

Ist das interne FF "Belegt-Halten" gesetzt, wird wieder der Zustand "Belegt-Halten" (HOLD) erreicht, der sich nur durch Aufhebung dieses Signales löschen läßt. Man benutzt ihn u.a. für den direkten Speicherzugriff und zum Auffrischen dynamischer Speicher.



3.2.5 Flußdiagramm der einzelnen T-Zyklen beim 8080

Entscheidung: Befehlsausführung beendet?

Bei nicht beendetem Befehl wird immer auf den T 1-Zyklus übergegangen. Das bedeutet, daß zunächst keine "Unterbrechung" (INT oder INTE) möglich ist.

Entscheidung: "Unterbrechung"

Liegt bis zum Ende einer Befehlsausführung eine Unterbrechungsanfrage vor, so wird das "Unterbrechungs"-Flip-Flop "INT-FF" gesetzt. Jetzt können Unterprogramme oder ergänzende Daten eingelesen werden. Die Unterbrechung "INTE" und "INT" löscht sich automatisch beim nächsten T 1-Zyklus.

Speicheradressenaufteilung

Es ist sinnvoll, die Adressen des Festwertspeichers mit ØØØØ (Hexadezimal) beginnen zu lassen, da nach jedem "RESET" der Programmzähler bei dieser Adresse anfängt. Sind bereits Programme vorhanden, müssen die Speicheradressen mit den im Programm festgelegten übereinstimmen.

Liegen keine Vorgaben fest, so wird eine Aufteilung nach *Abb. 3.2.6* vorgeschlagen. Nach den Festwertspeichern und einer Reserve für weitere Festwertspeicher wird mit der Adresse 2ØØØ (Hexadezimal) der Kellerspeicher (Stack-RAM) angeordnet. Er reicht bis zur Adresse 2Ø FF. Da der Stack-Pointer stets von der höheren Adresse nach unten zählt, kann bei einem Überlauf keine Information im Datenspeicher überschrieben werden, sondern nur eine leere Adresse oder eine ROM-Adresse angesprochen werden.

Ab Adresse 3ØØØ (Hexadezimal) sind speicheradressierte Ein- und Ausgänge untergebracht. Normalerweise liegen die Adressen für die Ein- und Ausgänge von Null bis 256 und werden sowohl über die Adressensammelschienenleitungen $A_0 - A_7$ als auch über $A_8 - A_{15}$ ausgegeben. Ab Adresse 4ØØØ (Hexadezimal) sind die RAM-Datenspeicher angeordnet.

Dieser Vorschlag hat den großen Vorteil, daß für die Minimalkonfiguration geschriebene Programme unverändert auch bei beliebigen Erweiterungen beibehalten werden können.

Erweiterungen

Programmspeicher

Reicht die Kapazität des Programmspeichers 8316 mit 2048 x 8 Bit nicht aus, so können weitere drei Speicher hinzugefügt werden *(Abb. 3.2.7)*, deren Auswahl über die Adressensammelschiene und die maskenprogrammierten "Chip-select"-Eingängen CS_1 — CS_3 erfolgt. Die Eingänge können bei der Bestellung so ausgewählt werden, daß sie entweder bei H- oder L-Signalen den Baustein freigeben. Welche Ausführung für CS_3 gewählt wird, hängt davon ab, ob das Status-Signal mit dem 8212 oder 8228 (siehe spätere Abb. 3.2.11) dekodiert wird.

Arbeitsspeicher (Datenspeicher)

Die Erweiterung der Arbeitsspeicher wird mit dem 1 aus 8 Dekoderbaustein 8205 vorgenommen *(Abb. 3.2.8)*. Es können an einen Baustein 8205 acht RAM-Speicher angeschlossen werden, die über die "Chip-select"-Eingänge $\overline{CE_1}$ freigegeben werden.

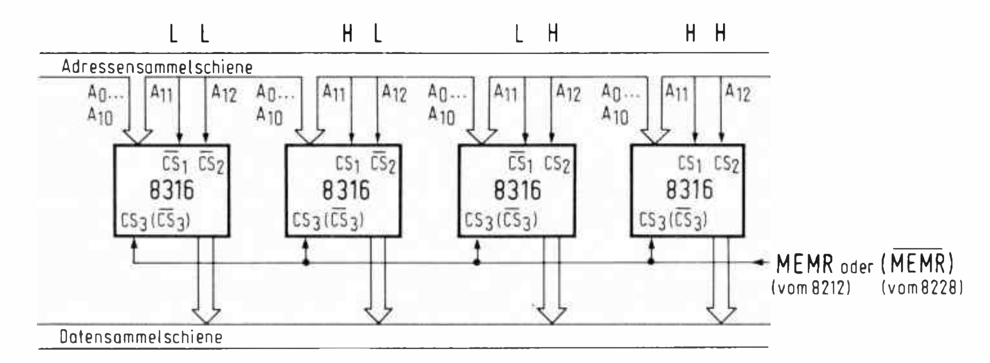
Ein- und Ausgänge

Ebenso kann die Zahl der Ein- und Ausgänge mit einem Baustein 8205 verachtfacht werden (*Abb. 3.2.9*). Sie werden paarweise über die Eingänge $\overline{DS_1}$ angewählt. Über die Eingänge DS_2 wird durch das Status-Signal bestimmt, ob eine Ein- oder Ausgabe stattfindet. Eine weitere Verknüpfung kann sinngemäß mit den Signalen DBIN und $\overline{WR}$ erfolgen.

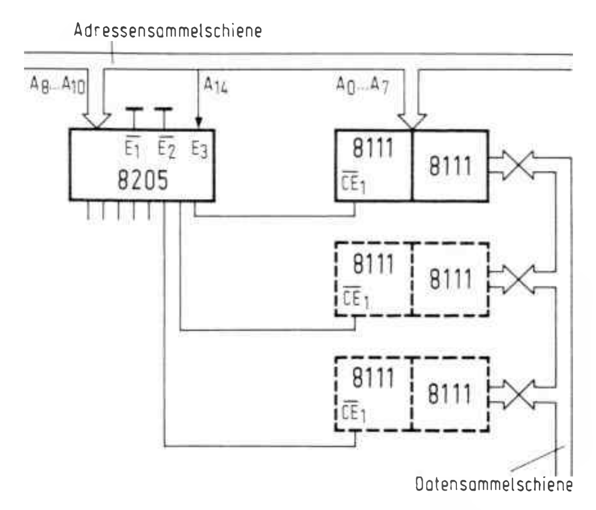
Reicht die mit einem Dekoder erreichte Erweiterungsmöglichkeit nicht aus, so kann unter Zuhilfenahme der drei "Chip-select"-Eingänge des Dekoderbausteines mit acht Dekodern eine Erweiterung auf das 64-fache erfolgen (*Abb. 3.2.10)*. Dies gilt sowohl für die Ein- und Ausgänge als auch für Speicherbausteine.

Beispiel mit Bausteinen höheren Integrationsgrades

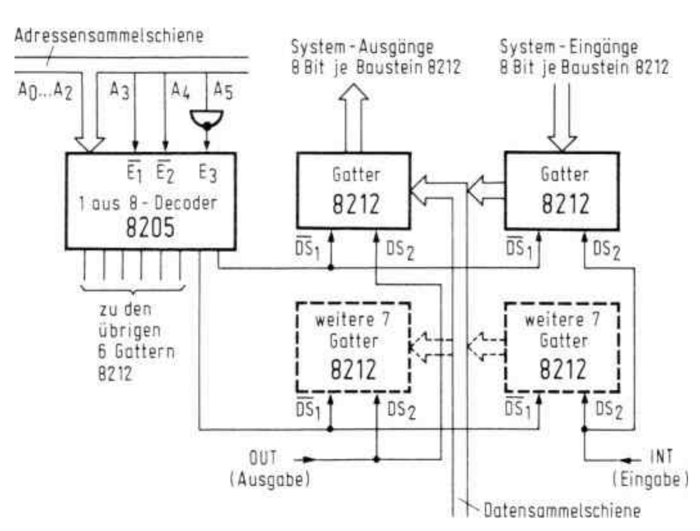
Ein wesentlich einfacherer Aufbau eines Systems ergibt sich mit den höher integrierten Bausteinen 8228 und 8255 *(Abb. 3.2.11)*. Das programmierbare Daten-Interface 8255 ersetzt von der Kapazität her drei Bausteine 8212. Da der Baustein 8255 aber sowohl die Ein- als auch die Ausgabe bewerkstelligen kann, werden insgesamt 6 Bausteine 8212 durch einen 8255 ersetzt.



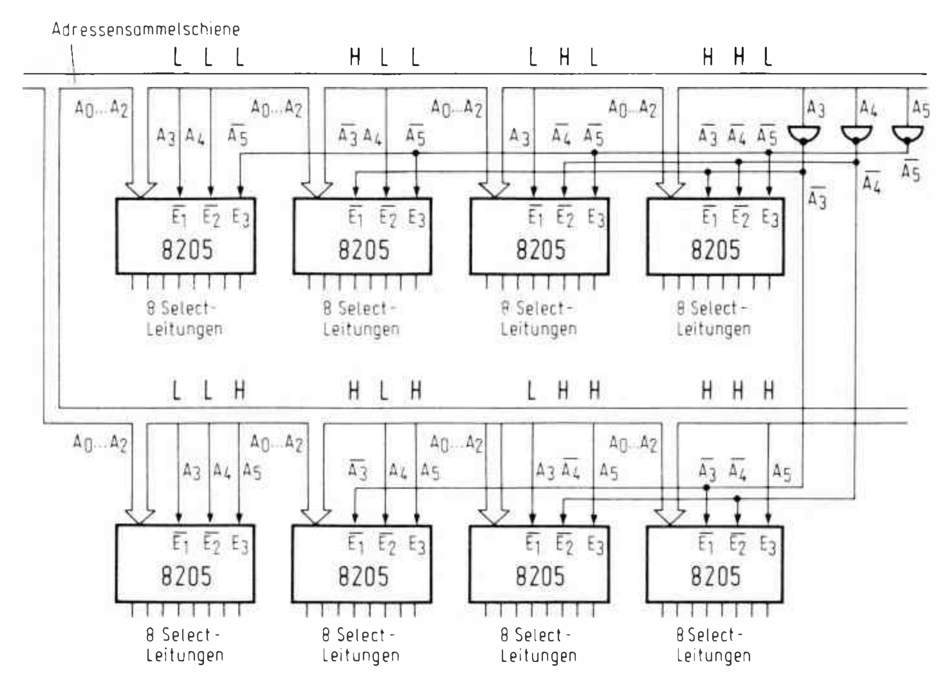
3.2.7 Adressendekodierung bei mehreren ROMs durch masken-kodierte Chipselect-Eingänge



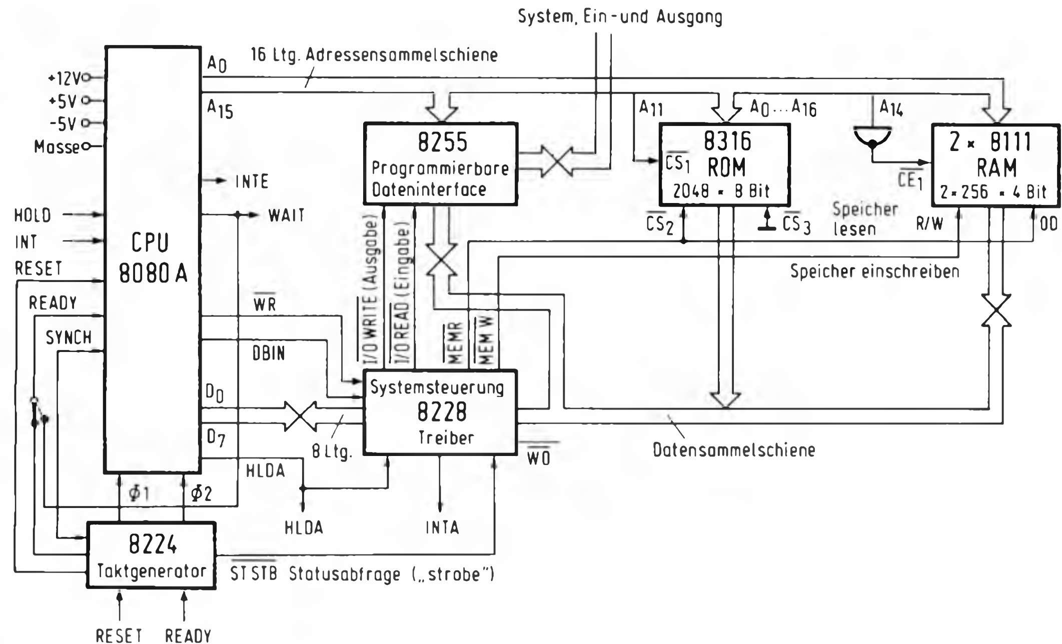
3.2.8 Adressendekodierung bei mehreren gleichen Bausteinen, hier am Beispiel der RAM-Datenspeicher



3.2.9 Erweiterung der System-Ein- und Ausgänge und Dekodierung der Adressen



3.2.10 Vorschlag für die Dekodierung von 64 Adressen, hier als Beispiel für 64 Ein-Ausgabeeinheiten



3.2.11 Minimalsystem eines Mikrocomputers mit dem Mikroprozessor 8080 und hochintegrierten Interface-Bausteinen

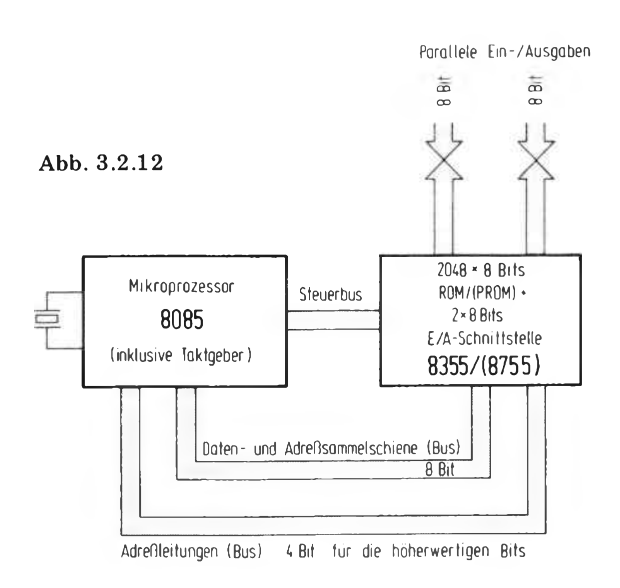
Der Systemsteuerungs- und Treiberbaustein 8228 ersetzt nicht nur den "STATUS-Latch" 8212, sondern macht über seinen Zweirichtungssammelschienentreiber die Datensammelschiene genügend niederohmig für den Anschluß mehrerer Speicher bzw. für die Verbindung der Datensammelschiene mit anderen Sub-Systemen.

Die Funktion des Lesespeichers ROM 8316 und Schreiblesespeichers 8111 ist die gleiche wie weiter oben beschrieben. Reichen die Speicherplätze nicht aus, so können mehrere Bausteine parallel geschaltet werden.

Sinngemäß kann auch das ganze System entsprechend den Anforderungen nach den vorne beschriebenen Schaltungen erweitert werden.

Weitere Vereinfachungen ergeben sich mit Bausteinen höheren Integrationsgrades. So kann mit dem Mikroprozessor 8085, einer Weiterentwicklung des 8080, mit zwei (Abb. 3.2.12) bis drei (Abb. 3.2.13) Bausteinen ein Mikrocomputersystem realisiert werden, je nachdem, ob externe Schreib-/Lesespeicher gebraucht werden oder die Register im Mikroprozessor ausreichen.

Abb. 3.2.12



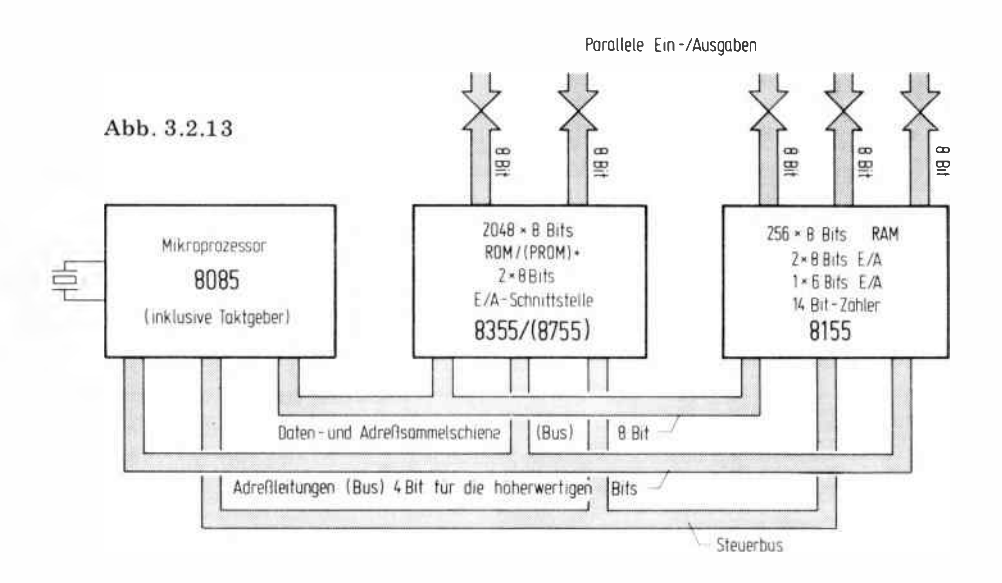

Parallele Ein-/Ausgaben
8 Bit
8 Bit
8 Bit
8 Bit
8 Bit
Mikroprozessor
8085
(inklusive Taktgeber)
2048 × 8 Bits
ROM/(PROM)+
2 × 8 Bits
E/A-Schnittstelle
8355/(8755)
256 × 8 Bits RAM
2 × 8 Bits E/A
1 × 6 Bits E/A
14 Bit-Zähler
8155
Daten- und Adreßsammelschiene (Bus) 8 Bit
Adreßleitungen (Bus) 4 Bit für die höherwertigen Bits
Steuerbus

Abb. 3.2.13

Der Mikroprozessor 8085 hat den gleichen Befehlssatz wie der Typ 8080, ergänzt durch zwei zusätzliche Befehle für Interrupt-Funktionen.

Für sehr einfache Anwendungsfälle reichen die 1-Chip-Mikrocomputer aus. Abb. 3.2.14 zeigt das Blockschaltbild des Types 8048 (Ausführung mit ROM)/8748 (Ausführung mit PROM).

Der Befehlssatz dieses Typs ist nicht kompatibel mit dem System 8080. Er enthält speziell auf das Anwendungsfeld zugeschnittene Befehle, die den ROM-Speicherbedarf verringern.

Die hier gezeigten Beispiele sollten dabei als Hilfe zum Verständnis des Mikroprozessors dienen, der in dem folgenden Abschnitt beschrieben wird.

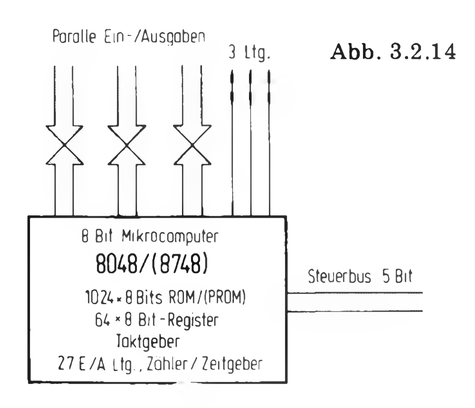


Abb. 3.2.14

4 Mikroprozessoren

4.1 Der Mikroprozessor 8008

Der 8008 ist eine in P-Kanal-MOS-Technologie aufgebaute 8 Bit-Zentraleinheit für das MCS-8 Mikrocomputersystem. Der Prozessor *(Abb. 4.1.1)* arbeitet über eine gemeinsame 8 Bit-Daten- und Adreß-Sammelschiene. Er hat zwei Eingangsanschlüsse "READY" und "INTERRUPT". Vier Ausgangsanschlüsse S_0, S_1, S_2 und SYNCH dienen zur Steuerung der verschiedenen Funktionen. Der Zeitmultiplex-Betrieb der Datensammelschiene gestattet die Übertragung der Steuerinformationen, einer 14 Bit-Adresse und der Daten zwischen der Zentraleinheit (CPU) und den externen Speichern.

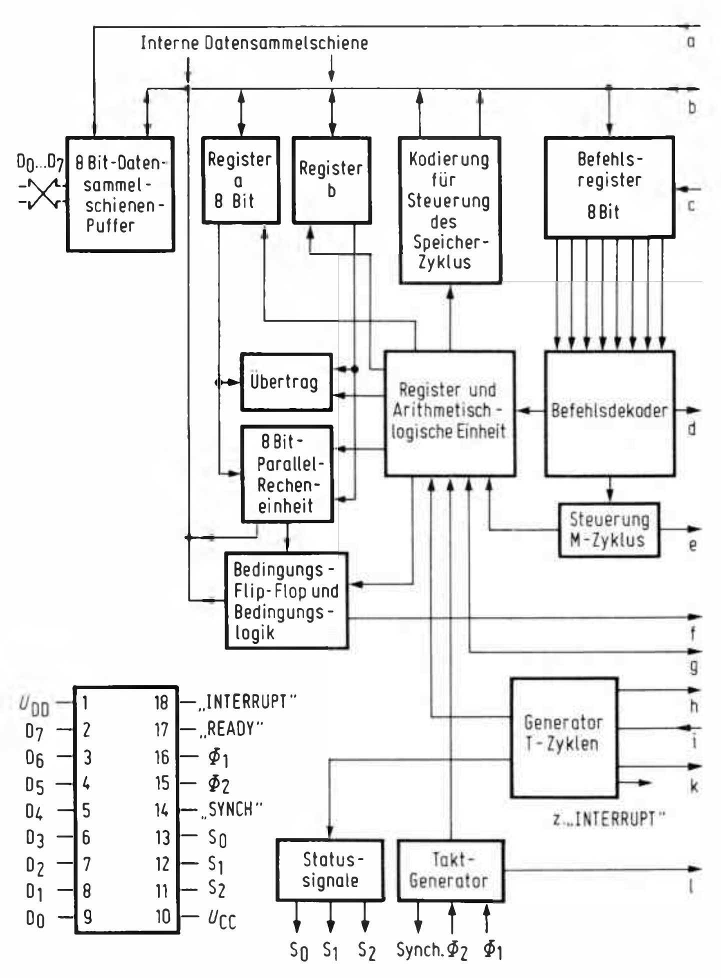
CPU

Die CPU enthält sechs 8 Bit-Datenregister, einen 8 Bit-Akkumulator, zwei 8 Bit-Kurzzeitregister, vier Kennzeichen-Bits für Bedingungen und eine 8 Bit-Parallel-Recheneinheit. Diese kann Additionen, Subtraktionen und logische Operationen ausführen.

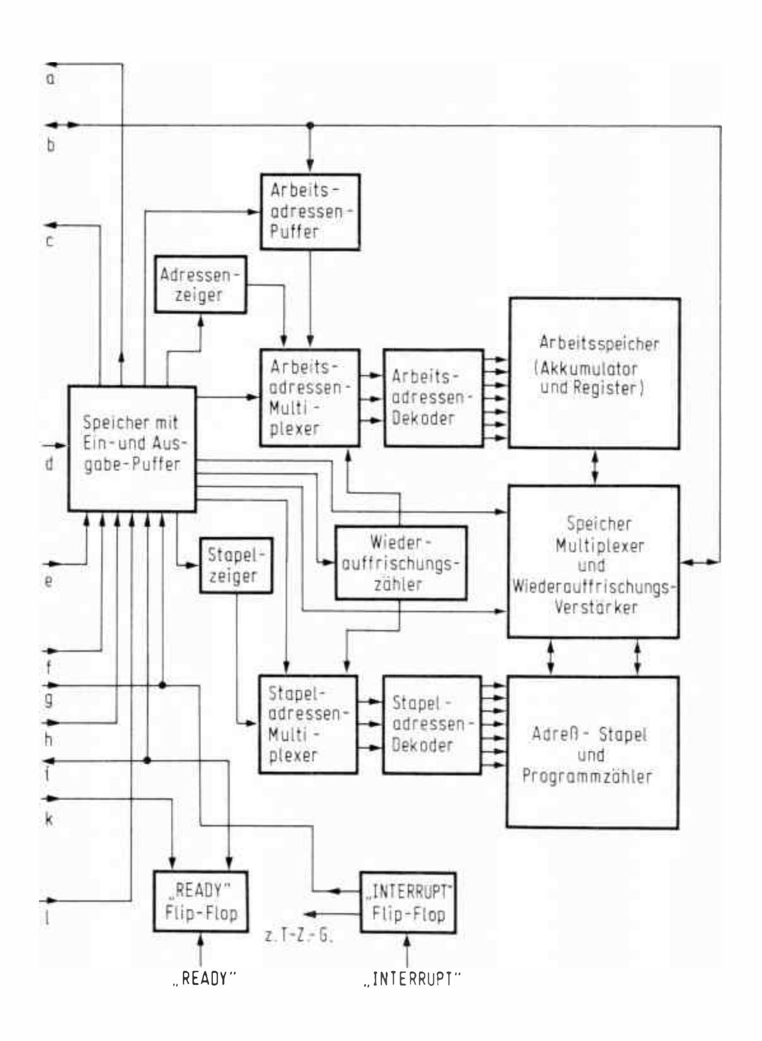
Ein interner Speicher enthält einen 14 Bit-Programmzähler und sieben 14 Bit-Worte. Er wird intern zum Speichern von Programm- und Unterprogrammadressen verwendet. Die 14 Bit-Adresse erlaubt die direkte Adressierung von 16 K-Speicherworten. Mit ihnen kann jede Mischung von RAM, ROM oder Schieberegister angesprochen werden.

Steuerteil

Der Steuerteil der CPU enthält eine Logik für die Speicherübertragung, arithmetische Kontrollen und logische Befehle. Die meisten Befehle sind in einem Byte (8 Bits) kodiert. Datenübertragungsbefehle benötigen zwei Bytes, Sprungbefehle drei Bytes. Die Zentraleinheit arbeitet mit 500 kHz. Befehle, die nicht den Speicher ansprechen, werden in 20 μs ausgeführt.



4.1.1 Blockschaltbild der CPU 8008



a
b
c
d
e
f
g
h
i
k
l
Arbeits-adressen-Puffer
Adressen-zeiger
Arbeits-adressen-Multi-plexer
Arbeits-adressen-Dekoder
Arbeitsspeicher (Akkumulator und Register)
Speicher mit Ein-und Aus-gabe-Puffer
Speicher Multiplexer und Wiederauffrischungs-Verstärker
Stapel-zeiger
Wieder-auffrischungs-zähler
Stapel-adressen-Multi-plexer
Stapel-adressen-Dekoder
Adreß- Stapel und Programmzähler
„READY" Flip-Flop
„INTERRUPT" Flip-Flop
z.T-Z.-G.
„READY"
„INTERRUPT"

zu 4.1.1

Ein- und Ausgänge

Alle Eingänge sind TTL-kompatibel, alle Ausgänge zur Ansteuerung von "low-power"-TTL geeignet.

Befehlsliste

Die Befehlsliste des 8008 besteht aus 48 Befehlen. Mit ihnen können Datenverarbeitung, binäre Arithmetik und Sprünge in Unterprogramme ausgeführt werden.

INTERRUPT-Anschluß

Der normale Programmablauf des 8008 kann durch den INTERRUPT-Anschluß unterbrochen werden. Das erlaubt die Bedienung von langsamen Eingabe/Ausgabe-Peripherie-Geräten, während das Hauptprogramm abläuft. Die "READY"-Leitung synchronisiert den 8008 mit dem Speicher-Zyklus.

Zeitablauf

Die vier Ausgänge "STATE" und "SYNCH" zeigen den jeweiligen Zeitablauf des Prozessors zu jedem Zeitpunkt während des M-Zyklus an.

Zeitablauf im Prozessor

Die interne Organisation der CPU gruppiert sich um eine interne 8 Bit Datensammelschiene. Alle Kommunikation innerhalb des Prozessors und mit externen Geräten geschieht über diese Schiene in Form von 8 Bits, seien es Adressen, Befehle oder Daten. Im gesamten MCS-8-System ist die logische Eins als hohes Potential "H" definiert, die logische Null als niedriges Potential "L".

Kodierung

Die T-Zyklus-Kodierung wurde bereits in Abschnitt 3.1 besprochen, ebenfalls der M-Zyklus. Er besteht typisch aus fünf T-Zyklen, wobei zwei zur Übermittlung der Adresse vom Speicher benützt werden, ein T-Zyklus für einen Befehl oder Datentransport und die beiden T-Zyklen T 4 und T 5 für die Ausführung des Befehles. Wenn der Prozessor mit einem langsamen Speicher betrieben wird, synchronisiert die "READY"-Leitung den Prozessor mit den Speichern. Wenn die Speicher nicht in

der Lage sind, Signale abzugeben oder Daten zu empfangen, geht der Prozessor in den "WAIT"-Zustand.

Impulsplan

Den Impulsplan für die verschiedenen T-Zyklen fanden Sie in Abb. 3.1.2. Der Empfang eines "INTERRUPT"-Signales wird durch Sprung nach T_{1I} quittiert (Fluß-Diagramm siehe Abb. 3.1.3). Wenn der Prozessor unterbrochen wird, ersetzt dieser Zyklus den Zyklus T 1. Nach einem "READY" folgt T 3. Der Zustand "STOPPED" zeigt an, daß ein HALT-Befehl empfangen wurde.

Viele Befehle des 8008 haben mehrere Zyklen und erfordern die beiden T-Zyklen T 4 und T 5 nicht. Dann werden diese Takte übersprungen und der 8008 arbeitet asynchron in Bezug auf die Zykluslänge. Zu erwähnen ist, daß der Zustand "WAIT" und der Zustand "STOPPED" in ihrer Länge undefinierbar sein können.

M-Zyklen

Der Aufbau der M-Zyklen erfolgt immer nach folgendem Schema:

> Im ersten M-Zyklus wird immer ein Befehl geholt (PCI).
> Der zweite und dritte M-Zyklus ist zum Lesen von Daten (PCR),
> zum Schreiben von Daten (PCW)
> oder für Eingabe/Ausgabe-Operationen (PCC) bestimmt.
> Die Art des Maschinen-Zyklus wird mit 2 Bits kodiert und auf der Datensammelschiene während T 2 übertragen.

Funktionsblöcke

Die vier grundlegenden Funktionsblöcke dieses Prozessors sind das Befehlsregister, der Speicher, die arithmetisch-logische Einheit und die Ein- und Ausgabepuffer. Sie verkehren untereinander über die interne 8 Bit-Datensammelschiene.

Befehlsregister

Das Befehlsregister ist das Herz aller Steuerungen im Prozessor. Befehle werden aus dem Speicher geholt, im Befehlsregister aufbewahrt und zur Steuerung des Speichers und der arithmetisch-logischen Einheit (ALU) dekodiert. Da die Ausführung

der Befehle nicht alle die gleiche Anzahl von T-Zyklen erfordert, steuert der Befehlsdekoder auch die einzelnen T-Zyklus-Durchgänge.

Interne Speicher

Der 8008 enthält zwei dynamische Speicher, den Rückstell-Adreßstapelspeicher und den Arbeitsspeicher. Diese internen Speicher werden automatisch bei jedem T 3-Zyklus aufgefrischt, aber auch im Zustand "WAIT" und "STOPPED". In jedem Falle werden sie bei jedem achten Takt komplett aufgefrischt.

Adressenstapel

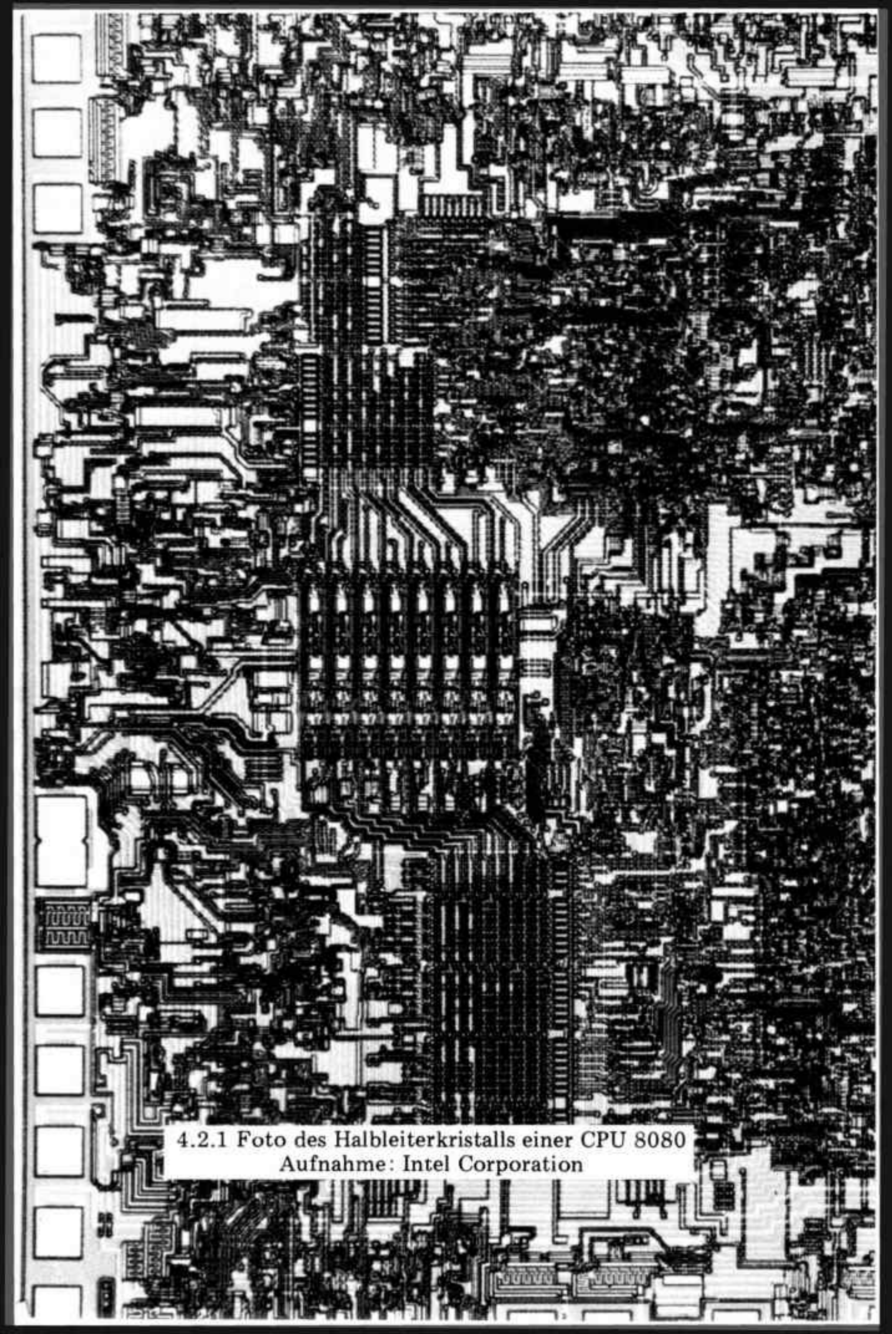
Der Adressenstapel enthält acht 14 Bit-Register. Ein Register wird als Programmzähler verwendet. Die anderen sieben erlauben die Abstellung von Adressen für Unterprogramme. Im Stapel wird automatisch der Inhalt des Programmzählers nach der Ausführung eines "RETURNs" wieder in den Programmzähler überschrieben. Die Stapelregister werden als zuletzt eingelesene und zuerst ausgelesene Rückstellspeicher verwendet. Eine 3 Bit-Adressenanzeige gibt an, welches der acht Register gerade als Programmzähler verwendet wird. Wenn die maximale Kapazität des Stapels überschritten wird, stellt sich die Adressenanzeige zurück und der Inhalt des "untersten" Registers wird zerstört. Der Programmzähler wird sofort nach dem Aussenden der Adreß-Bits um eins erhöht.

Programmzähler

Der 14 Bit-Programmzähler erlaubt die direkte Adressierung von 16 k-Bytes im Speicher. Durch die Hernahme eines Eingabe/Ausgabe-Befehles für eine Umschaltung kann der Speicher erweitert werden.

Arbeitsspeicher

Der Arbeitsspeicher enthält den Akkumulator (Register A) und sechs zusätzliche 8 Bit-Register B bis L. Alle arithmetischen Operationen gebrauchen den Akkumulator als einen der Operanten. Alle Register sind unabhängig und können für kurzfristige Speicherungen verwendet werden. Im Falle von Befehlen, die eine Arbeit mit einem Register im externen Speicher erfor-



4.2.1 Foto des Halbleiterkristalls einer CPU 8080
Aufnahme: Intel Corporation

dern, erlauben die Index-Register "H" und "L" eine indirekte Adressierung.

Arithmetik

Alle arithmetischen und logischen Operationen wie Addieren, Addieren mit Übertrag, Subtrahieren, Subtrahieren mit Übertrag, Exklusives ODER, Vergleich, Inkrement und Dekrement, werden in der 8 Bit parallel-arithmetischen Einheit ausgeführt, die eine vorausschauende Einheit für den Übertrag enthält.

Kurzzeitregister

Zwei Kurzzeitregister "a" und "b" werden verwendet, um den Inhalt des Akkumulators und den Operanten für die ALU-Operationen aufzubewahren. Zusätzlich werden sie für kurzfristige Adreßaufbewahrung und Datenaufbewahrung während interner Prozessorumschichtungen verwendet. Vier Steuerbits, Flip-Flops für Übertrag, für Null, für Vorzeichen und für Paarigkeit werden als Ergebnis jeder arithmetischen oder logischen Operation gesetzt. Diese Bits erlauben bedingte Verzweigungen durch "CALL", "JUMP" oder "RETURN" bei eventuellen Bedienungsfehlern.

Eingabe-Ausgabepuffer

Der Ein-Ausgabe-Puffer ist die einzige Verbindung zwischen dem Prozessor und dem restlichen System. Jeder der acht Puffer kann nach beiden Richtungen übertragen und wird sowohl durch das Befehlsregister als auch durch die Zeitkontrolle gesteuert. Der Puffer ist in der Lage, ausgangsseitig "LOW-POWER"-TTL zu treiben, und ist eingangsseitig an die TTL-Pegel angepaßt.

4.2 Der Mikroprozessor 8080

Der von Intel entwickelte Mikroprozessor 8080 ist einer der universell einsetzbaren Prozessoren. Im Gegensatz zum Typ 8008, welcher in P-MOS-Technologie hergestellt wird, arbeitet der 8080 mit der schnelleren N-MOS-Technologie. *Abb. 4.2.1*

zeigt ein Chip-Foto des 8080. Die Additionszeit im Register für ein Datenwort verkleinert sich damit von 20 auf 2 µs. Gleichzeitig besitzt der 8080 eine Adressenkapazität von 64 K-Bytes.

CPU

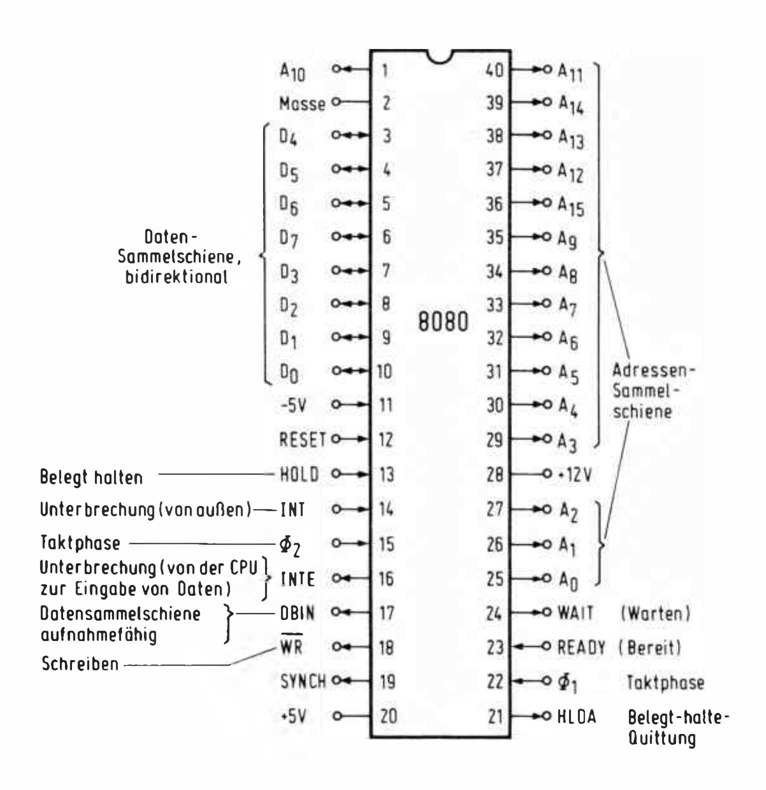
Der Mikroprozessor ist in einem Gehäuse mit 40 Anschlüssen untergebracht *(Abb. 4.2.2)*. Er besitzt sechs 8 Bit-Universalregister, drei Kurzzeitregister und einen Akkumulator. Die Register können entweder individuell oder paarweise adressiert werden. Arithmetische und logische Befehle steuern vier testbare Kennzeichen-Flip-Flops. Ein fünftes Kennzeichen ist für dezimalarithmetische Operationen vorgesehen. Der 8080 hat einen 16 Bit-Stapelzeiger (''STACK-POINTER''), wobei der Speicher das zuletzt Eingelesene zuerst ausgibt. Die 16 Bit-Stapel-Adresse steuert die Adressierung. Dieser Stapel gibt dem 8080 die Möglichkeit, mehrfach eine Prioritätsunterbrechung zu handhaben, in dem der Prozessorinhalt schnell gespeichert oder restauriert werden kann. Es erlaubt ebenso das fast unbegrenzte Einschachteln von Unterprogrammen.

Systemaufbau

Besonderer Wert beim Entwurf war auf die Vereinfachung des Systems gelegt worden. Getrennte 16gleisige Adressen- und 8gleisige Datenschienen (letztere in beiden Richtungen) machen die Verbindungen zum Speicher und der Ein/Ausgabe flexibel und schnell. Wenn gewünscht, können 512 Ein- und Ausgangskanäle direkt angesprochen werden. Die Signale zur Steuerung der Interface zum Speicher und zu den Ein- und Ausgängen werden direkt im Mikroprozessor erzeugt. Eine zusätzliche Steuerung der Adressen- und der Datensammelschienen geschieht mit dem ''Besetzt halte''-Signal (HOLD). Es sieht die Möglichkeit vor, die Prozessoroperation zu sperren und die Adressen- und Datensammelschienen in einen hochohmigen Zustand zu versetzen. Dies erlaubt, diese Sammelschienen mit anderen Steuereinheiten für einen direkten Speichereingriff oder Vielfach-Prozessor-Operation zu verbinden.

Ein 16 Bit-Programmzähler gestattet die direkte Adressierung von Speichern bis zu 64 k-Bytes. Der Mikroprozessor ist für Dezimal-, Binär- und Double precision-Arithmetik geeignet.

Der Prozessor ist in der Lage, Unterbrechungen für Daten mit erhöhter Priorität einfließen zu lassen.



4.2.2 Anschlüsse des 8080

Datensammelschiene

In *Abb. 4.2.3* finden Sie das Blockschaltbild der CPU 8080.

Die Datensammelschiene $D_0 - D_7$ (Pinzuordnung siehe Abb. 4.2.2) gestattet einen Verkehr in beiden Richtungen zwischen der CPU, den Speichern und den Ein- und Ausgabeeinheiten für Befehle und Datenfluß. D_0 hat die kleinste Wertigkeit. Eine Pufferstufe mit einem elektronischen Umschalter für den Datenfluß von innen nach außen oder umgekehrt, verbindet die äußere Datensammelschiene mit der internen Schiene. An der internen Schiene sind der Akkumulator, das Befehlsregister, ein Multiplexer, die Kurzzeitregister und die Kennzeichen-Flip-Flops angeschlossen.

Befehlsregister

Das Befehlsregister führt auf den Befehlsdekoder mit dem M-Zyklusgeber.

Zeitgeber und Steuerung

Es geht weiter zum Zeitgeber mit Steuereinheit. Zu diesem Block führen 12 Verbindungen mit denen folgende Signale ein- bzw. ausgegeben werden:

Steuer-Ein- und Ausgänge

"RESET"-Eingang

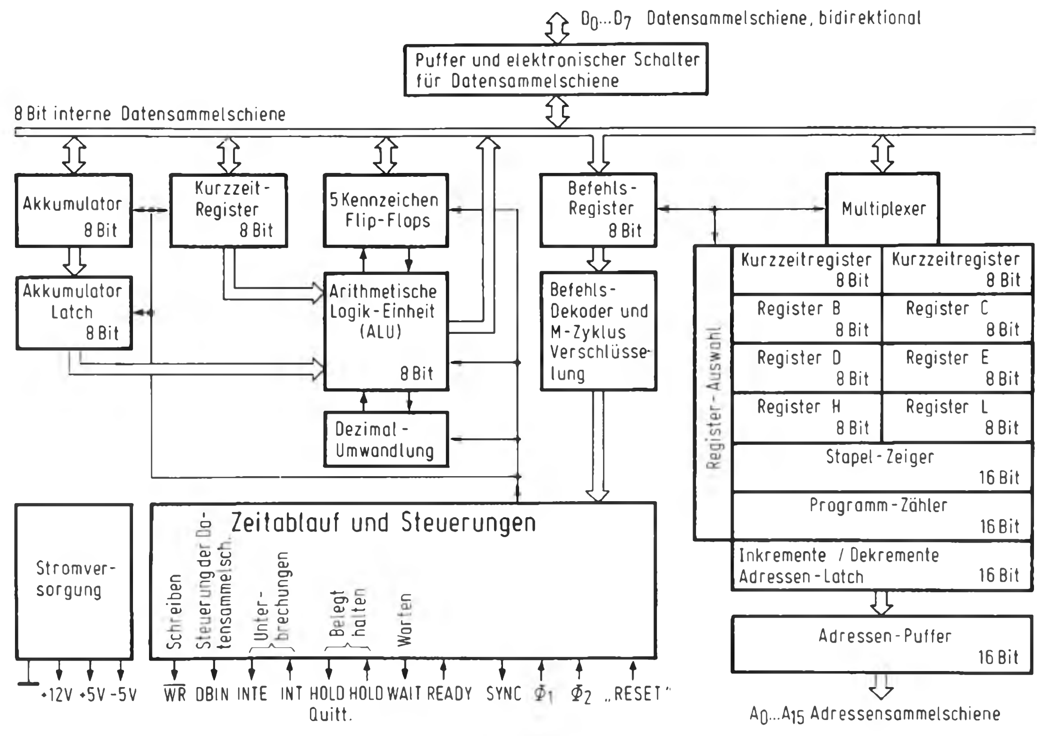
Beim Anlegen des "RESET"-Signales wird der Inhalt des Programmzählers gelöscht und das Befehlsregister auf Null gesetzt. Nach diesem "RESET"-Vorgang wird das neue Programm beginnend beim Platz Null im Speicher gestartet. Die "INTE" und "HLDA"-Flip-Flops werden ebenso rückgesetzt. Zu beobachten ist, daß die Kennzeichen, der Akkumulator und die Register nicht gelöscht werden.

Takt-Eingänge

An die Eingänge Ø1 und Ø2 werden die Taktsignale angelegt.

"READY"-Eingang

Ein "READY"-Signal zeigt dem Mikroprozessor an, daß gültige Daten aus dem Speicher oder vom Eingang für die Daten-



4.2.3 Blockschaltbild der CPU 8080

schiene verfügbar sind. Dieses Signal wird gebraucht, um die CPU mit langsameren Speichern oder langsameren Ein- und Ausgabeeinheiten zu synchronisieren. Wenn eine Adresse ausgesendet wird und der 8080 kein "READY"-Signal empfängt, geht die CPU in den Wartezustand, bis das "READY"-Signal eintrifft.

"SYNCH"-Ausgang

Am "SYNCH"-Ausgang steht ein Signal zur Verfügung, welches identisch mit den T-Zyklen ist.

"Warte"(WAIT)-Ausgang

Das Warte-Signal bestätigt, daß die CPU im Wartezustand ist.

"HOLD"-Eingang

Das Belegt-Halte-Signal veranlaßt die CPU, in den Haltezustand zu gehen. Dieses Signal erlaubt z.B. die Wiederauffrischung externer Speicher über Adressen- und Datensammelschiene vorzunehmen, sobald der Mikroprozessor die Benutzung der Sammelschiene für den laufenden Maschinen-Zyklus beendet hat. Es hält die Adressen- und Datensammelschienen für andere Signale "belegt". Dies wird unter folgenden Bedingungen akzeptiert:

1. Die CPU ist im "Halt"-Zustand
2. Die CPU ist im T_2 oder T_W-Zyklus und das "READY"-Signal ist aktiviert.

Beim Erreichen des "HOLD"-Zustandes werden die Adressensammelschienen A_0 bis A_{15} und die Datensammelschiene D_0 bis D_7 hochohmig. Die CPU quittiert diesen Zustand mit dem "HLDA"-Signal am Anschluß 21.

"HLDA"-Ausgang

Dieses Signal erscheint nach dem "HOLD"-Signal und zeigt an, daß die Daten- und Adreßsammelschiene hochohmig ist. Das "HLDA"-Signal beginnt bei:

1. T_3 für Speicherlesen oder Eingabe
2. der Taktperiode, die T_3 für einschreiben in den Speicher oder Ausgabe folgt.
   Zur Beachtung: Das "HLDA"-Signal erscheint nach dem Anstieg von dem Takt Ø1 und die Hochohmigkeit erfolgt erst nach dem Anstieg von Ø2.

"INT"-Eingang

Interrupt request, das bedeutet Unterbrechungsanfrage. Die CPU reagiert auf eine Unterbrechungsanfrage an diesem Eingang erst am Ende der Befehlsausführung oder während eines Haltens. Wenn die CPU in dem "Besetzt"-Zustand ist oder wenn der "INTERRUPT"-Flip-Flop rückgesetzt ist, wird diese Anfrage nicht berücksichtigt.

"INTE"-Ausgang

Dieser Ausgang zeigt den Zustand des internen Unterbrechungs-Flip-Flops an. Dieses Flip-Flop kann entweder durch "Enable" oder "Disable" INTERRUPT-Befehle gesetzt bzw. rückgesetzt werden.

"DBIN"-Ausgang

Data bus in, d.h. die Sammelschiene ist auf Eingang geschaltet. Das "DBIN"-Signal zeigt den externen Einheiten an, daß die Sammelschiene empfangsbereit ist. Dieses Signal wird benutzt, um die Daten vom Speicher oder der Eingabeeinheit in die CPU zu transferieren.

"WRITE"-Ausgang ($\overline{WR}$)

Das Schreibe-Signal wird gebraucht, wenn in die Speicher Signale eingeschrieben oder an die Ausgabeeinheit gegeben werden. Die Daten auf der Sammelschiene sind verfügbar, während das $\overline{WR}$-Signal aktiviert ist.

Register

Von der internen Datensammelschiene wird ein Multiplexer gespeist, der den Registern zugeordnet ist. Der Multiplexer steuert u.A. auch die Registerauswahl. Den Registern ist ferner ein "Inkrementer/Dekrementer" zugeordnet und ein Adressen-"Latch", welches die Adressen über eine Pufferstufe an die 16 Bit-Parallel-Adressensammelschiene ausgibt.

Adressensammelschiene

Diese Adressensammelschiene kann bis zu 64 k-Bytes Speicheradressen adressieren und Ein- und Ausgabeeinheiten mit je 256 Ein- und 256 Ausgabekanäle ansprechen.

Akkumulator, ALU und Hilfsstufen

Der Akkumulator arbeitet mit der arithmetisch logischen Einheit (ALU) eng zusammen. An dieser arithmetisch logischen Einheit hängen Hilfsstufen wie die Kennzeichen-Flip-Flops und die Dezimalkorrektur.

Spannungs- und Taktversorgung

Gegen Masse wird die Spannung U_{DD} + 12 Volt, U_{CC} + 5 Volt, U_{BB} — 5 Volt benötigt. Alle Spannungen müssen die Tolerenz von ± 5 % einhalten. Die beiden Taktphasen Ø1 und Ø2 sind nicht TTL-kompatibel.

Befehlssatz

Die Akkumulator-Gruppenbefehle schließen arithmetische und logische Operatoren mit direkter, indirekter und unmittelbarer Adressierung ein. "MOVE", "LOAD" und "STORE"-Befehlsgruppen (Verschiebe-, Lade- oder Speicher) geben die Möglichkeit, entweder acht- oder 16 Bit-Daten zwischen dem Speicher, den sechs Arbeitsregistern und den Akkumulator zu verschieben. Auch hier können wieder direkte oder indirekte Adressen verwandt werden.

Die Möglichkeit, zu verschiedenen Teilen des Programmes zu verzweigen, wird mit "JUMP", "JUMP CONDITIONAL" und "COMPUTED JUMP" erreicht. Ebenso ist die Möglichkeit, Unterprogramme aufzurufen oder von Unterprogrammen zurückzukehren sowohl "bedingt" als auch automatisch vorgesehen. Der "RESTART" ist für unterbrochene Vektoroperationen.

Die Fähigkeit, den Speicherinhalb zu inkrementieren oder dekrementieren ist bei sechs normalen Registern und dem Akkumulator vorhanden. Eine weitere Programmiermöglichkeit ist durch die Fähigkeit des Rotierens des Akkumulators links- oder rechtsherum erschlossen. Die Eingabe und Ausgabe kann entweder über speicheradressierte Ein- und Ausgangs-Kanäle oder direkt adressierte Ein- und Ausgänge mit dem 8080 Befehlsspektrum vorgenommen werden.

Folgende Spezialbefehlsgruppen vervollständigen den 8080 Befehlskatalog: Der Nichtoperationsbefehl: HALT, um die Prozessorausführung zu stoppen und der DAA-Befehl, welcher das Arbeiten mit Dezimalzahlen erlaubt. STC setzt das Übertrags-

Bit direkt, der CMC-Befehl gestattet die Komplementbildung, CMA komplementiert den Inhalt des Akkumulators und XCHG tauscht den Inhalt der beiden 16 Bit-Registerpaare (D, E) und (H, L) direkt aus.

4.3 Auswahlkriterien bei der Beurteilung von Mikroprozessoren

Außer den beiden in den Abschnitten 3.1, 3.2, 4.1 und 4.2 besprochenen Mikroprozessoren 8008 und 8080, die von Intel entwickelt und von mehreren anderen Firmen als "second source" geliefert werden, gibt es eine beachtliche Anzahl von Mikroprozessoren anderer Hersteller, die teils lieferbar, teils als Muster und nur teils als Entwurf oder Vorschlag vorliegen.

Wenn auch die kommerziellen Gründe, wie Lieferfähigkeit und Zweitlieferant häufig bei der Auswahl maßgebend sind, will der Verfasser nachstehend vorwiegend technische Qualitätsmerkmale als Kriterium heranziehen.

4.3.1 Technologie

Mikroprozessoren werden heute in

P-Kanal- N-Kanal- C-	MOS	Technologie
TTL- ECL- I^2L-	(Bipolar)	

hergestellt.

4.3.1.1 P-Kanal-MOS-Technologie

Die P-Kanal-MOS-Technologie, zuerst als Standard- oder Hochvolttechnik bekannt geworden *(Abb. 4.3.1.1)*, ist die heute am meisten in der Herstelung von integrierten MOS-Schaltungen verwendete Technik. Sie erfordert relativ hohe Schwellen- bzw. Versorgungsspannungen. Auf Grund minimaler Design- und Herstellungsschritte eignet sie sich sehr gut für die Großintegration (LSI).

Sehr bald wurde diese Technologie bei Halbleiterspeichern durch die Silicon-Gate-Technik ergänzt. Sie unterscheidet sich im Material der Steuerelektrode *(Abb. 4.3.1.2)* aus polykristallinem Silizium statt aus Aluminium. Da das Gate als dritte Leiterbahnebene verwendet werden kann, sind größere Packungsdichten möglich. Die Technik ist TTL-kompatibel, meist mit Versorgungsspannungen von + 5 V und − 12 V. Silicon-Gate ist eine selbstjustierende Technik mit vielen Unterschieden im Herstellungsprozess im Vergleich zur Standardtechnik.

Noch bessere Kompatibilität mit TTL ergibt die Depletion-Load-Technik mit doppelter Ionenimplantation *(Abb. 4.3.1.3)*. Integrierte MOS-Schaltungen in dieser Technik erfordern nur eine Versorgungsspannung von + 5 V.

Durch gleichzeitige Verwendung von leitenden und nichtleitenden Transistoren entstehen Schaltungen, die schnell arbeiten und eine kleine Verlustleistung haben.

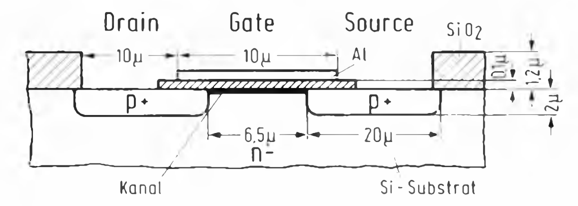
Vorteile der P-Kanal-Technologie sind:

Eingefahrene Fertigungstechnik, daher
große Fertigungsausbeuten,
geringer Preis.

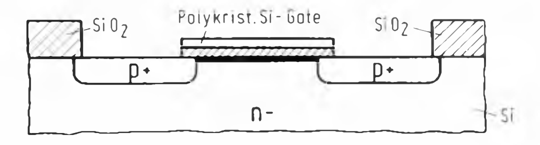
Nachteilig ist die langsamere Arbeitsgeschwindigkeit, die aber bei vielen Anwendungen keine Rolle spielt.

In P-Kanal-MOS-Technologie ausgeführte Mikroprozessoren sind:

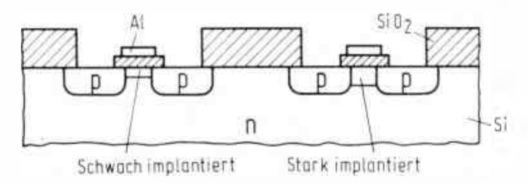
Intel 4004
Intel 4040
Intel 8008
Fairchild PPS-25
Rockwell PPS-4
Rockwell PPS-8
National Semiconductors PACE
National Semiconductors IMP-4
National Semiconductors IMP-8
National Semiconductors IMP-16
National-Semiconductors SC/MP (alt)
Burroughs Mini-D
Mosteck MK 5065
Digital M 7341
Texas TMS 1000 NC
Siemens SAB 4004
Siemens SAB 4040



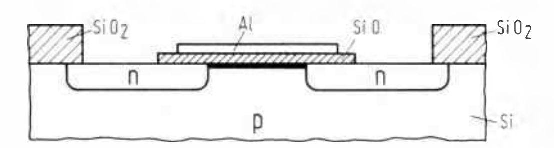
4.3.1.1 Aufbau eines P-Kanal-MOS-Transistors



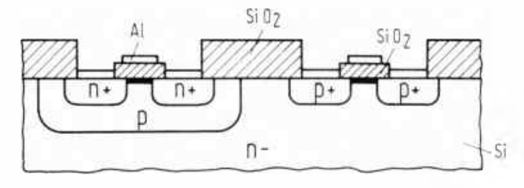
4.3.1.2 Aufbau eines MOS-Transistors in Silicon-Gate-Technik



4.3.1.3 Aufbau eines MOS-Transistors in Depletion-Load-Technik



4.3.1.4 Aufbau eines N-Kanal-MOS-Transistors in Standard-Technik



4.3.1.5 Aufbau eines C-MOS-Transistors

4.3.1.2 N-Kanal-Technologie

Die in N-Kanal-Technologie ausgeführten Mikroprozessoren bezeichnet man auch als 2. Generation. In *Abb. 4.3.1.4* finden Sie den Dotierungsaufbau dieser Technik. Die Transistoren haben unterschiedliche Dotierungen. Source und Drain sind N-Diffusionen, das Substrat ist P-dotiertes Silizium. Bei der Herstellung sind sehr strenge Anforderungen an die Sauberkeit des Oxids zu stellen. N-Kanal-Technik kann entweder mit Silicon-Gate-Technik oder mit Depletion-Load-Technik kombiniert werden. Durch die dreimal bessere Leitfähigkeit des Kanals ergibt sich eine wesentliche Erhöhung der Schaltgeschwindigkeit und eine dreimal größere Packungsdichte. In N-Kanal-MOS-Technologie ausgeführte Mikroprozessoren sind:

Intel 8080	Siemens SAB 8080
Intel 8085	Siemens SAB 8085
Intel 8048/8748	Siemens SAB 8048/8748
Motorola 6800	
Fairchild F 8	
Signetics 2650	
Toshiba TLCS-12	
NEC µPD 756 D	
GI CP 1600	
MOS Technology 6501/6502/6503	
National Semiconductors SC/MP (neu)	

4.3.1.3 C-MOS-Technik

Die C-MOS-Technik hat prinzipiell ein anderes Herstellungsverfahren *(Abb. 4.3.1.5).* Ein großer Vorteil sind die niedrigen Verlustleistungen (10^{-5} gegenüber TTL) und der große Versorgungsspannungsbereich (3 ... 15 V). Nachteilig ist die relativ geringe Packungsdichte und die dadurch schlechtere Wirtschaftlichkeit. Die C-MOS-Technik bekannt gewordenen Mikroprozessoren sind:

RCA COSMAC (2 chips)
Solid State Scientific 7001
Intersil IM 6100

4.3.1.4 TTL-Technik

Mit bipolaren integrierten Schaltungen lassen sich wesentlich höhere Arbeitsgeschwindigkeiten realisieren. Jedoch ist es wegen der geringeren Packungsdichte nicht mehr möglich, die CPU auf einem Chip unterzubringen, obwohl moderne Schottky-Dioden-TTL-Techniken hinsichtlich der Packungsdichte gegenüber Standard-TTL bereits erheblich besser ist. Mit Schottky-Dioden lassen sich die gleichen Arbeitsgeschwindigkeiten wie mit Standard-TTL bei wesentlich geringerem Leistungsbedarf realisieren. Folgende Mikroprozessoren sind in TTL-Technik angeführt:

Intel 3000
Fairchild 9400-System
MMI 5701/6701
Advanced Micro Devices 2901

4.3.1.5 ECL-Technik

Mit integrierten Schaltungen in ECL-Technik lassen sich die Arbeitsgeschwindigkeiten nochmals um den Faktor 2 verbessern, man erreicht eine Zykluszeit von 55 ns. Der einzige in dieser Technik angekündigte Mikroprozessor ist die
Motorola MC 10800-Serie.

4.3.1.6 I^2L-Technik

Die I^2L-Technik läßt die höchste Packungsdichte bei den bipolaren Schaltungen zu. In dieser Technik bietet
Texas SBP 0400
und Transitron
je einen Mikroprozessor an.

4.3.2 Wortlänge

Die Wortlänge gibt an, wieviel Bits im Prozessor parallel verarbeitet werden können. Obwohl man zwar nicht generell sagen kann, je größer die Wortlänge, desto besser, ist doch den möglichen Aufgaben eine Mindestwortlänge zugeordnet, damit die Datenverarbeitung nicht zu kompliziert wird und die Rechenzeit nicht zu lange dauert.

4.3.2.1 4-Bit-Parallelverarbeitung

4 Bit genügen im allgemeinen für folgende Anwendungen:

BCD-Display-Steuerungen
BCD-Rechenoperationen
Elektronische Kassenregister
Büromaschinen
Buchhaltungsmaschinen
Kredit-Karten-Prüfung
Hilfsmittel für intelligente Meßgeräte
Spielautomaten

Der Vorteil liegt in den geringen Kosten und in der leichten Handhabung solcher einfachen Mikroprozessoren wie:

Intel 4004
Intel 4040
Fairchild PPS-25
National Semiconductor IMP-4
Rockwell PPS-4
Teledyne TDK-52A
Siemens SAB 4004
Siemens SAB 4040

4.3.2.2 8-Bit-Parallelverarbeitung

Der 8-Bit-Datenfluß im Mikroprozessor gestattet den Aufbau ausgewachsener Regeleinrichtungen und Prozesskontrollen:

Numerische Maschinensteuerung
Verkehrs- und Ampelsteuerungen
Intelligente Terminals
Hochwertige automatisierte Meßeinrichtungen
Datenbündler
Daten-Ein- und Ausgabeeinrichtungen
Dezentralisierte Datenvorverarbeitung
Bordrechner für Flugzeuge und Autos
Schreibautomaten
Kleinrechner
Medizinische Diagnose und Auswerf-Systeme
Ersatz von Minicomputern durch Mikrocomputer

Der Vorteil der größeren Wortlänge macht sich in der erhöhten Schnelligkeit, Adressierbarkeit mit ASCII- oder EBCDIC-Code, in der vereinfachten Ein- und Ausgabe und einfacherer

Programmierbarkeit bemerkbar. Typische 8-Bit-Mikroprozessoren sind:

Intel 8008
Intel 8080
Motorola M 6800
National Semiconductor IMP-8
Signetics 2650
RCA COSMAC
DIGITAL MPS PDP-8A
Rockwell PPS-8
Siemens SAB 8080

4.3.2.3 12-Bit-Parallelverarbeitung

Für viele Anwendungsfälle sind 8-Bit Wortlänge zu gering, werden 16 Bit jedoch nicht benötigt. Diese Lücke wird durch die Mikroprozessoren von

Intersil ISD-8
Toshiba TLCS 12
General Automation LSI 12

geschlossen.

4.3.2.4 16-Bit-Parallelverarbeitung

Mikroprozessoren mit 16-Bit-Datenwortlänge sind bereits universell einsetzbar, da auch Minicomputer mit diesem Format arbeiten. 16-Bit-Mikroprozessoren werden von den Herstellern unter:

Intel 3000 (9 Slices)
National Semiconductor IMP 16,C,L,P
Computer Automation M/LSI 1
Raytheon RP 16
General Automation LSI 16
Teledyne TDK 52 B
Texas SBP 0400 (4 Slices)
NC μPD 756 D

angeboten.

4.3.2.5 Erweiterungsmöglichkeit der Wortlänge

Bei einigen Mikroprozessoren gibt es die Möglichkeit, durch Verwendung mehrerer CPU's die Wortlänge aufzustocken. Die Chips müssen von vorn herein dafür konzipiert sein. Dies ist vorwiegend bei den Ausführungen in bipolarer Technologie der Fall.

4.3.3 Weitere Kriterien bei der Beurteilung von Mikroprozessoren

Je nach Anwendungsfall können auch andere Eigenschaften bei der Auswahl des geeigneten Mikroprozessors wichtig sein.

Adressenkapazität

Sie reicht bis auf kleinere mehr kundenspezifische Prozessoren von 4 ... 64 K-Bytes.

Intel 4004	4 K Bytes
Intel 4040	4 K Bytes
Intel 8008	16 K Bytes
Intel 8080	64 K Bytes
Intel 8085	64 K Bytes
Motorola 6800	64 K Bytes
National Semiconductor	64 K Bytes
Siemens SAB 4004	4 K Bytes
Siemens SAB 4040	4 K Bytes
Siemens SAB 8080	64 K Bytes
Siemens SAB 8085	64 K Bytes

Adressierarten

Getrennte Adressensammelschiene

Sie erlaubt ein schnelleres Arbeiten, da die Datensammelschiene nicht zur Adressenausgabe mitbenutzt zu werden braucht. Außerdem spart man sich die externen Zwischenspeicher. Die Mikroprozessoren
Intel 8080, Motorola 6800 und Siemens SAB 8080
arbeiten mit getrennten Adressensammelschienen.

Arbeitsgeschwindigkeit

Sie ergibt sich aus der Technologie und Organisation des ganzen Mikrocomputersystems. Bei Prozessoren in MOS-Technologie liegt die Additionszeit pro Datenwort etwa zwischen 2...60 μs.

Intel 8008	20 μs
Intel 8080	2 μs

Prozessoren in bipolarer Technologie erreichen etwa eine Zehnerpotenz geringere Zeiten.

Unterbrechungsmöglichkeit für Unterprogramme; Stapelgröße für Adressenspeicherung während der Unterprogramme

Diese Anforderung ist je nach Einsatzbereich sehr unterschiedlich.

Intel 4004	4 x 12 Bits
Intel 8008	8 x 14 Bits
Intel 8080	unbegrenzt durch externes RAM
Intel 8085	unbegrenzt durch externes RAM
Motorola 6800	unbegrenzt durch externes RAM
National Semiconductor IMP 16	16 x 16 Bits
Siemens SAB 8080	unbegrenzt durch externes RAM
Siemens SAB 8085	unbegrenzt durch externes RAM

Anzahl der Register; verfügbare Bausteine zur Ergänzung

Von großer Wichtigkeit ist die Verfügbarkeit der zum Aufbau des gewünschten Systems benötigten Bausteine, Programme und Hilfseinrichtungen wie:

Speicher   Ein- und Ausgabebausteine   Taktgenerator.

Befehlskatalog

Der Befehlskatalog der Mikroprozessoren beträgt

Intel 4004, Siemens SAB 4004	45 Befehle
Intel 4040, Siemens SAB 4040	60 Befehle
Intel 8008	48 Befehle
Intel 8080, Siemens SAB 8080	78 Befehle
Intel 8085, Siemens SAB 8085	80 Befehle
Intel 8048/8047	65 Befehle
Motorola 6800	72 Befehle

Wobei die 48 Befehle des 8008 in den 78 des 8080 kompatibel enthalten sind.

Hilfsmittel

Folgende Hilfsmittel werden zur Komplettierung eines Mikrocomputersystems benötigt:

Entwicklungs-System	PROM-Programmiereinrichtung
System Monitor	Simulator
MACRO-Assembler	Cross-Compiler
Text Editor	

Sie werden im allgemeinen von den Mikroprozessor-Herstellern mit angeboten und gehören sozusagen zum Kundendienst *(Abb. 4.3.3)*.



4.3.3 Hilfsmittel zur Mikrocomputerentwicklung. In der Mitte: Mikrocomputer als Steckkarte. Aufnahme: Intel Corporation

5 Programmierung

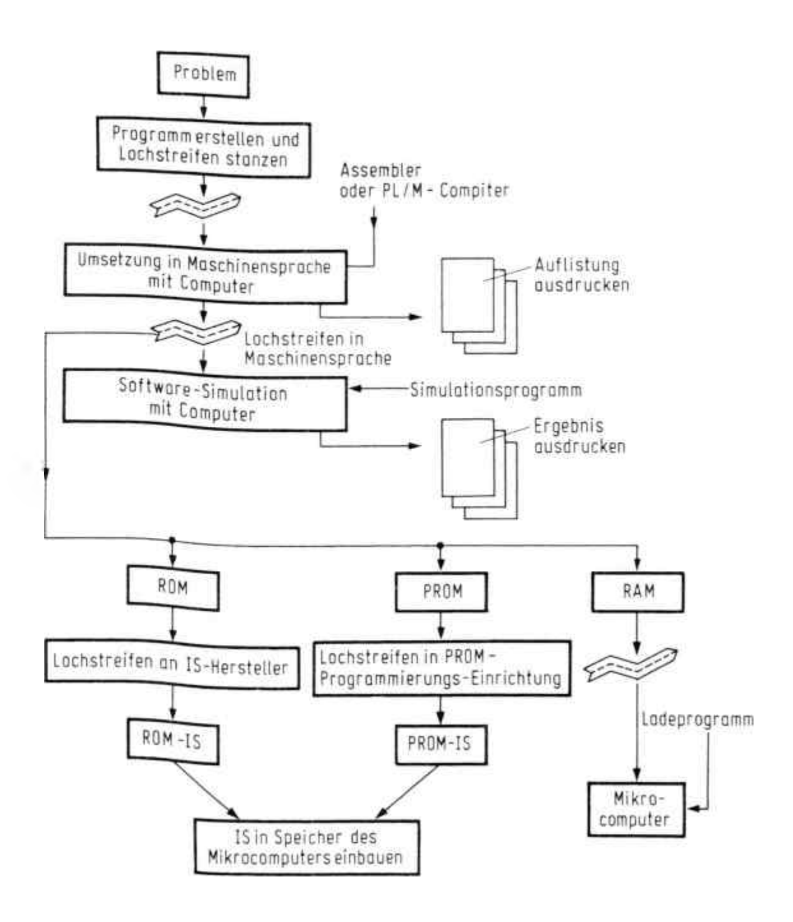
Programmieren heißt, dem Computer die Befehle geben, die er braucht, um eine Verarbeitung, z.B. eine Rechnung, eine Sortierung oder eine andere Aufgabe ausführen zu können. Dazu muß eine Programmiersprache benutzt werden, die der Computer auch versteht. Ist das Programm in einer dem Rechner nicht verständlichen Sprache geschrieben, so muß es mittels eines "Compilers" übersetzt werden *(Abb. 5.0.1)*. Die Programmiersprachen lassen sich in zwei Gruppen aufteilen.

5.1 Problemorientierte Programmiersprachen

Diese problemorientierten Sprachen werden auch höhere Programmiersprachen genannt, zu ihnen gehören

ALGOL
FORTRAN
COBOL
BASIC
PL/M

Diese Sprachen haben einen engen Bezug zur wissenschaftlichen Darstellung, sind sehr übersichtlich und einfach zu erstellen. Für die Mikrocomputersysteme Intel 8008 und 8080 sowie für Motorola MC 6800 gibt es Compiler, die PL/M in Maschinenkode direkt umsetzen. Andernfalls kann man diese Übersetzung mit einem speziellen Programm durch eine EDV-Anlage vornehmen.



5.0.1 Programmierung von Mikrocomputern

5.2 Maschinenorientierte Programmiersprache

Assembler ist die rein maschinenorientierte Programmiersprache. Es besteht ein enger Bezug zur Maschinensprache, die nur aus Binärzahlen besteht. Programme in Assembler-Sprache laufen sehr schnell und benötigen wenige Speicherzellen. Für jeden Maschinentyp ist die Assembler-Sprache unterschiedlich, sie wird als Betriebsanleitung mitgegeben. Ein Assembler-Programm setzt dann die Programmiersprache in die eigentliche Maschinensprache (Binärzahlen) um. Diese Umsetzung kann mit einer EDV-Anlage vorgenommen werden. Der Assembler kann aber auch auf einem 8008 oder 8080-System ablaufen.

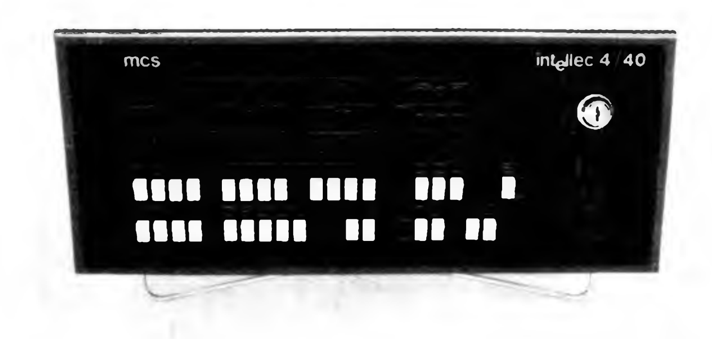
Der Assembler testet die eingegebenen Programme auf formale Fehler, liefert einen Lochstreifen in Maschinenkode und eine Auflistung des Programmes.

5.3 Programmieren der PROMs bzw. ROMs

Der Maschinenkode-Lochstreifen kann zum Programmieren der ROMs und PROMs verwendet werden. ROMs werden durch Anfertigung einer dem Lochstreifen entsprechenden Maske vom Halbleiterhersteller direkt programmiert. Für die Übertragung des Lochstreifens auf PROMs gibt es dafür speziell entwickelte Programmiergeräte *(Abb. 5.3)*.

5.4 Laden von RAMs

Ist man sich bei der Entwicklung des Programms noch nicht sicher, ob dieses alle gewünschten Funktionen erfüllt, kann beim 8080-System der Maschinenkode-Lochstreifen auch mit einem Ladeprogramm direkt in den RAM-Speicher oder in einen an die Stelle des ROMs gesteckten RAM-Speicher geladen werden. Letzteres wird man immer dann tun, wenn ein Programm nur einmalig benutzt werden soll. Abb. 5.0.1 veranschaulicht bis dahin diesen Vorgang übersichtlich.



5.3 Mikrocomputerentwicklungssystem Intellec 4/40, rechts die Fassung zur Programmierung des PROM.
Aufnahme: Intel Corporation

5.5 Befehlslisten für Mikroprozessoren

Wie geht es dann weiter in den Mikroprozessor? Der eine Weg ist, daß der Mikroprozessor eine gewisse Anzahl fester Befehle verarbeiten kann:

Intel 4040, Siemens SAB 4040	60 Befehle
Intel 8008	48 Befehle
Intel 8080, Siemens SAB 8080	78 Befehle
Intel 8085, Siemens SAB 8085	80 Befehle
Motorola 6800	72 Befehle

Diese Befehle müssen dem Mikroprozessor über die Datensammelschiene von dem Programmspeicher (ROM, PROM oder RAM, siehe Abschn. 5.3, 5.4 und Abb. 5.0.1) erteilt werden. Befehle, die sich nicht im oben genannten Befehlsvorrat befinden, kann der Mikroprozessor nicht ausführen. Andererseits werden für viele Anwendungen nicht alle Befehle benötigt. Um Ihnen einen Eindruck einer solchen Befehlsliste zu geben, ist in *Tabelle 5.0* die Originalbefehlsliste des 8080 in englischer Spra-

Tabelle 5.0: SILICON GATE MOS 8080, INSTRUCTION SET (Summary of Processor Instructions

Mnemonic	Description	Instruction Code [1] D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Clock [2] Cycles
$MOV_{r1,r2}$	Move register to register	0	1	D	D	D	S	S	S	5
MOV M, r	Move register to memory	0	1	1	1	0	S	S	S	7
MOV r, M	Move memory to register	0	1	D	D	D	1	1	0	7
HLT	Halt	0	1	1	1	0	1	1	0	7
MVI r	Move immediate register	0	0	D	D	D	1	1	0	7
MVI M	Move immediate memory	0	0	1	1	0	1	1	0	10
INR r	Increment register	0	0	D	D	D	1	0	0	5
DCR r	Decrement register	0	0	D	D	D	1	0	1	5
INR M	Increment memory	0	0	1	1	0	1	0	0	10
DCR M	Decrement memory	0	0	1	1	0	1	0	1	10
ADD r	Add register to A	1	0	0	0	0	S	S	S	4
ADC r	Add register to A with carry	1	0	0	0	1	S	S	S	4
SUB r	Subtract register from A with borrow	1	0	0	1	1	S	S	S	4
ANA r	And register with A	1	0	1	0	0	S	S	S	4
XRA r	Exclusive Or register with A	1	0	1	0	1	S	S	S	4
ORA r	Or register with A	1	0	1	1	0	S	S	S	4
CMP r	Compare register with A	1	0	1	1	1	S	S	S	4
ADD M	Add memory to A	1	0	0	0	0	1	1	0	7
ADC M	Add memory to A with carry	1	0	0	0	1	1	1	0	7
SUB M	Subtract memory from A	1	0	0	1	0	1	1	0	7
SBB M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	7
ANA M	And memory with A	1	0	1	0	0	1	1	0	7
XRA M	Exclusive Or memory with A	1	0	1	0	1	1	1	0	7
ORA M	Or memory with A	1	0	1	1	0	1	1	0	7

Mnemonic	Description	Instruction Code [1] D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Clock [2] Cycles
CMP M	Compare memory with A	1	0	1	1	1	1	1	0	7
ADI	Add immediate to A	1	1	0	0	0	1	1	0	7
ACI	Add immediate to A with carry	1	1	0	0	1	1	1	0	7
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	7
SBI	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7
ANI	And immediate with A	1	1	1	0	0	1	1	0	7
XRI	Exclusive Or immediate with A	1	1	1	0	1	1	1	0	7
ORI	Or immediate with A	1	1	1	1	0	1	1	0	7
CPI	Compare immediate with A	1	1	1	1	1	1	1	0	7
RLC	Rotate A left	0	0	0	0	0	1	1	1	4
RRC	Rotate A right	0	0	0	0	1	1	1	1	4
RAL	Rotate A left through carry	0	0	0	1	0	1	1	1	4
RAR	Rotate A right through carry	0	0	0	1	1	1	1	1	4
JMP	Jump unconditional	1	1	0	0	0	0	1	1	10
JC	Jump on carry	1	1	0	1	1	0	1	0	10
JNC	Jump on no carry	1	1	0	1	0	0	1	0	10
JZ	Jump on zero	1	1	0	0	1	0	1	0	10
JNZ	Jump on no zero	1	1	0	0	0	0	1	0	10
JP	Jump on positive	1	1	1	1	0	0	1	0	10
JM	Jump on minus	1	1	1	1	1	0	1	0	10
JPE	Jump on parity even	1	1	1	0	1	0	1	0	10
JPO	Jump on parity odd	1	1	1	0	0	0	1	0	10
CALL	Call unconditional	1	1	0	0	1	1	0	1	17
CC	Call on carry	1	1	0	1	1	1	0	0	11/17
CNC	Call on no carry	1	1	0	1	0	1	0	0	11/16
CZ	Call on zero	1	1	0	0	1	1	0	0	11/17
CNZ	Call on no zero	1	1	0	0	0	1	0	0	11/17
CP	Call on positive	1	1	1	1	0	1	0	0	11/17
CM	Call on minus	1	1	1	1	1	1	0	0	11/17
CPE	Call on parity even	1	1	1	0	1	1	0	0	11/17
CPO	Call on parity odd	1	1	1	0	0	1	0	0	11/17

Mnemonic	Description	Instruction Code [1]								Clock [2]
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Cycles
RET	Return	1	1	0	0	1	0	0	1	10
RC	Return on carry	1	1	0	1	1	0	0	0	5/11
RNC	Return on no carry	1	1	0	1	0	0	0	0	5/11
RZ	Return on zero	1	1	0	0	1	0	0	0	5/11
RNZ	Return on no zero	1	1	0	0	0	0	0	0	5/11
RP	Return on positive	1	1	1	1	0	0	0	0	5/11
RM	Return on minus	1	1	1	1	1	0	0	0	5/11
RPE	Return on parity even	1	1	1	0	1	0	0	0	5/11
RPO	Return on parity odd	1	1	1	0	0	0	0	0	5/11
RST	Restart	1	1	A	A	A	1	1	1	11
IN	Input	1	1	0	1	1	0	1	1	10
OUT	Output	1	1	0	1	0	0	1	1	10
LXI B	Load immediate register Pair B & C	0	0	0	0	0	0	0	1	10
LXI D	Load immediate register Pair D & E	0	0	0	1	0	0	0	1	10
LXI H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	10
LXI SP	Load immediate stack pointer	0	0	1	1	0	0	0	1	10
PUSH B	Push register Pair B & C on stack	1	1	0	0	0	1	0	1	11
PUSH D	Push register Pair D & E on stack	1	1	0	1	0	1	0	1	11
PUSH H	Push register Pair H & L on stack	1	1	1	0	0	1	0	1	11
PUSH PSW	Push A and Flags on stack	1	1	1	1	0	1	0	1	11
POP B	Pop register pair B & C off stack	1	1	0	0	0	0	0	1	10
POP D	Pop register pair D & E off stack	1	1	0	1	0	0	0	1	10
POP H	Pop register pair H & L off stack	1	1	1	0	0	0	0	1	10
POP PSW	Pop A and Flags off stack	1	1	1	1	0	0	0	1	10
STA	Store A direct	0	0	1	1	0	0	1	0	13
LDA	Load A direct	0	0	1	1	1	0	1	0	13
XCHG	Exchange D & E, H & L, Registers	1	1	1	0	1	0	1	1	4

Mnemonic	Description	Instruction Code [1] D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Clock Cycles [2]
XTHL	Exchange top of stack, H & L	1	1	1	0	0	0	1	1	18
SPHL	H & L to stack pointer	1	1	1	1	1	0	0	1	5
PCHL	H & L to program counter	1	1	1	0	1	0	0	1	5
DAD B	Add B & C to H & L	0	0	0	0	1	0	0	1	10
DAD D	Add D & E to H & L	0	0	0	1	1	0	0	1	10
DAD H	Add H & L to H & L	0	0	1	0	1	0	0	1	10
DAD SP	Add stack pointer to H & L	0	0	1	1	1	0	0	1	10
STAX B	Store A indirect	0	0	0	0	0	0	1	0	7
STAX D	Store A indirect	0	0	0	1	0	0	1	0	7
LDAX B	Load A indirect	0	0	0	0	1	0	1	0	7
LDAX D	Load A indirect	0	0	0	1	1	0	1	0	7
INX B	Increment B & C registers	0	0	0	0	0	0	1	1	5
INX D	Increment D & E registers	0	0	0	1	0	0	1	1	5
INX H	Increment H & L registers	0	0	1	0	0	0	1	1	5
INX SP	Increment stack pointer	0	0	1	1	0	0	1	1	5
DCX B	Decrement B & C	0	0	0	0	1	0	1	1	5
DCX D	Decrement D & E	0	0	0	1	1	0	1	1	5
DCX H	Decrement H & L	0	0	1	0	1	0	1	1	5
DCX SP	Decrement stack pointer	0	0	1	1	1	0	1	1	5
CMA	Compliment A	0	0	1	0	1	1	1	1	4
STC	Set carry	0	0	1	1	0	1	1	1	4
CMC	Compliment carry	0	0	1	1	1	1	1	1	4
DAA	Decimal adjust A	0	0	1	0	0	1	1	1	4
SHLD	Store H & L direct	0	0	1	0	0	0	1	0	16
LHLD	Load H & L direct	0	0	1	0	1	0	1	0	16
EI	Enable Interrupts	1	1	1	1	1	0	1	1	4
DI	Disable interrupt	1	1	1	1	0	0	1	1	4
NOP	No-operation	0	0	0	0	0	0	0	0	4

Notes: 1. DDS or SSS — 000 B — 001 C — 010 D — 011 E — 100 H — 101 L — 110 Memory — 111 A.
2. Two possible cycle times, (5/11) indicate instruction cycles dependent on condition flags.



5.7 Mikrocomputer(entwicklungs)-System in Betrieb. Von links nach rechts: Drucker, Universal PROM-Programmierungseinrichtung, Bedienungsteil mit Datensichtgerät, Hauptkonsole des Mikrocomputerentwicklungsgerätes, Modul, der an der Datensammelschiene angeschlossen ist und Plattenspeicher. Aufnahme: Intel Corporation

che abgedruckt. Es soll in diesem Buch nicht näher darauf eingegangen werden. Nur einige Bemerkungen:

0 entspricht dem "L" nach DIN Norm 41 785 Blatt 4
1 entspricht dem "H" nach DIN Norm 41 785 Blatt 4

Auf der linken Seite sind die Befehle in mnemonischer Kode, die sich zur Gedächtnishilfe für den Programmierer leichter merken läßt. Daneben die Beschreibung des Befehles. Dann folgt die Binärzahl, die die Programmspeicher auf die Datensammelschiene geben. Rechts ist die Anzahl der benötigten T-Zyklen für die Übermittlung angegeben.

5.6 Mikroprogrammierung

Der andere Weg ist die Mikroprogrammierung, wie sie beispielsweise beim Intel 3000-System angewendet wird. Hier muß der Anwender bis ins letzte Detail über die interne Beschaltung des Prozessors Bescheid wissen. Die erforderliche Entwicklungskapazität dafür ist beträchtlich.

5.7 Abschätzung der erforderlichen Programmierzeiten

Nachfolgend soll Ihnen nur eine grobe Abschätzung des Zeitaufwandes für die Software gegeben werden. Ein geübter Programmierer kann etwa 5-7 Befehle pro Stunde ausarbeiten. Die Anzahl zur Darstellung einer Funktion mit einem Mikroprozessor hängt stark von seiner Wortlänge ab. Ein vollständiges Programm setzt sich gewöhnlich aus mehreren hundert Befehlen zusammen.

Die gleiche Zeit, wie die Erstellung des Programmes, erfordert noch einmal das "Debugging", das heißt die Fehlersuche und Beseitigung von formellen und logischen Fehlern. Die Zeit erscheint sehr hoch, wird aber auch immer wieder von erfahrenen Spezialisten aus der Datenverarbeitung bestätigt.

Abb. 5.7 zeigt ein komplettes Mikrocomputer(entwicklungs)-System in Betrieb.

6 Wie ersetzt man eine fest verdrahtete Logik durch ein Mikrocomputersystem?

Diese Frage wird am häufigsten gestellt. Zuerst sehen wir uns die Gemeinsamkeiten beider Systeme an: Beide Systeme haben Eingänge und Ausgänge.

6.1 Eingänge beim Mikrocomputersystem

Ähnlich, wie bei fest verdrahteter Logik können die Eingänge entweder nur 1 Bit haben, also nur eine Aussage ja/nein beinhalten oder aus mehreren Binärzahlen bestehen, mit denen man Meßwerte oder alphanumerische Zeichen übermitteln kann. Je nach der Anzahl der Stellen kann z.B. das ganze Alphabet sowie Zahlen beliebiger Größe dual dargestellt werden.

Am Beispiel einer Ampelsteuerung können die Eingänge sowohl Druckknöpfe für Fußgängerübergänge, Informationssignal von Bahnschranken, Lichtschranken oder Induktionsschleifen, die den Verkehrsfluß der einzelnen Zufahrtsstraßen zählen, als auch zeitliche Startsignale der vorhergehenden Ampelgruppen für die sogenannte grüne Welle bei bestimmten Geschwindigkeiten sein.

6.2 Ausgänge beim Mikrocomputersystem

Auch hier gibt es Ausgänge mit nur 1 Bit für eine reine ja/nein-Aussage, wie auch zu 4 oder 8 Bit zusammengefaßte Ausgangseinheiten, die alphanumerische Zeichen übertragen und über einen weiteren Dekoder Einzelfunktionen auslösen.

Um beim Beispiel der Ampelsteuerung zu bleiben, werden entweder 1 Bit oder die dekodierten Ausgänge über Schaltglieder, wie Optokoppler und Thyristoren, mit den verschiedenen Lampen grün, orange und rot sowie Informationsleitungen zu den nächsten Ampelgruppen verbunden.

6.3 Was übernimmt das Mikrocomputersystem?

Es übernimmt mittels eines geschriebenen und, wie im Abschnitt 5.3 und Abb. 5.0.1 in den Mikrocomputer eingebrachten Programmes, alle Funktionen, die früher von Relais oder später von einer TTL-Logik ausgeführt wurden. Änderungen sind durch Auswechseln des ROMs oder PROMs in Sekunden zu erledigen, wo früher ein Monteur vielleicht Tage oder Wochen für die Neubeschaltung gebraucht hat.

7 Anwendungsgebiete

Generell kann gesagt werden, daß der Mikroprozessor (im Mikrocomputer) überall dort einsetzbar ist, wo bisher eine fest verdrahtete Logikschaltung Verwendung fand. Der Einsatz lohnt sich meist nicht, wenn sehr einfache übersichtliche Problemstellungen vorliegen.

Bei schnellen, gleichbleibenden Verarbeitungen muß von Fall zu Fall verglichen werden. Besonders günstig sind die Anwendungen bei Problemen:

hoher Komplexität,
wechselnden Eingangsdaten,
zu erwartenden Änderungen der Datenverarbeitung,
gleichzeitiger Einsatz des Grundgerätes oder des Types für verschiedene Probleme
für gemischte Operationen, bei denen Rechenoperationen häufig sind.

In dem Abschnitt 9 sind bereits viele Anwendungsgebiete im Detail angesprochen. Hier noch einmal einige Beispiele

Bordcomputer:
- in Autos
- in Flugzeugen
- in Schiffen
- für militärischen Einsatz

Datenverarbeitung:
- Bankanlagen,
- Datenkonzentratoren
- Datenvorverarbeitung (übernahme einfacher Probleme durch den Mikrocomputer)
- Emulation von Kleinrechnern
- Fakturiergeräte
- Multiprocessing
- Programmierbare Taschenrechner

DV-Peripheriegeräte:
- Bandgeräte
- Bildschirmgeräte
- Drucker
- Fernschreiber
- Floppy-Disk
- Intelligente Terminals
- Kartenleser
- Klarschriftleser

Industrieanwendungen:
- Aufzugsteuerung
- Automatische Bonder
- Dezentralisierte Regelsysteme
- Manipulatoren
- NC-Maschinen-Steuerung
- Pressensteuerung
- Produktionskontrollen
- Prozeßkontrollen
- Sortierautomaten
- Verkehrsampeln
- Verkehrsleitsysteme

Konsumgüter:
- Datenzeile in Fernsehgeräten
- Fahrkartenautomaten
- Kabelfernseher
- Münzwechsler
- Rechnende Waagen
- Registrierkassen
- Spielautomaten
- Spielzeug
- Zapfsäulenabrechnung

Meßgeräte:
- Bauelementeprüfautomaten
- Bedienungssichere Meßgeräte
- Oszillografen
- Radargeräte
- Rechnende Meßgeräte
- Spektrum Analyser

Medizinische Elektronik:
- Diagnosegeräte
- Laborauswertgeräte
- Patientenüberwachung

Nachrichtentechnik:
- Automatische Kanalüberwachung
- Vermittlungsämter

Physikalische Geräte:
- Meßgrößenumwandler
- Reaktor-Überwachung

8 Schlußbetrachtung

Frage: Was ist ein Mikroprozessor?

Antwort: Ein Mikroprozessor ist die CPU, also ein Bestandteil eines Mikrocomputers. Als integrierter Standardbaustein enthält er Rechen- und Leitwerk, deren interner Ablauf in der Regel von außen mit einem (programmierbaren) Festwertspeicher beeinflußbar ist. Der Mikroprozessor besteht entweder aus einem Chip oder wird bei den schnelleren bipolaren Typen aus mehreren Chips, den sogenannten Prozessorelementen (slices) zusammengesetzt.

Frage: Was ist ein Mikrocomputer?

Antwort: Ein Computer mit Mikroprozessor. Anwendungsgebiete von Mikrocomputern:

Ersatz von nicht ausgelasteten Minicomputern und festverdrahteter Logik. Die Wirtschaftlichkeitsgrenze liegt zwischen 30 bis 90 Logikbausteinen (MSI) und hängt weitgehend von den auszuführenden Operationen ab. Einsatz z.B. bei der Steuerung von Prozessen und Maschienen, für die Überwachung von Anlagen und Geräten, bei der Vermittlung von Nachrichten- bzw. Datenverbindungen.

Bei der heute erreichten Packungsdichte moderner Technologien ist ein Mikrocomputer in seiner Leistungsfähigkeit vergleichbar mit einem Minicomputer von vor einigen Jahren.

Ein Verzeichnis mit über 300 Literaturstellen finden Sie in dem Buch „Von der Schaltalgebra zum Mikroprozessor"

Franzis-Verlag

---

10 Fachausdrücke – Glossar

Access: Zugriff, Zugang z.B. zu einer Speicherstelle

Acknowledge: Quittung

A/D-converter: Analog/Digital-Wandler

Adder: Addierer, Addierwerk

Adresse: Ein bestimmtes Wort zur Kennzeichnung eines Speicherplatzes, eines Speicherbereiches oder einer Funktionseinheit

Adreßbus: Die Mikroprozessoren der zweiten Generation haben eine getrennte Sammelschiene für die Adressen. Dies erlaubt ein schnelleres Arbeiten, da die Datensammelschiene nicht zur Adressenausgabe mitbenutzt zu werden braucht.

Adressenkapazität: Anzahl der vom Mikroprozessor ansprechbaren Adressen von Speichern und Eingabe-Ausgabe-Bausteinen. Mikroprozessoren der zweiten Generation haben heute 64 kByte.

Address pointer: Adressenzeiger

Adreßregister: ein zur Aufnahme von Adressen bestimmter Zwischenspeicher

Address-stack: Adreßstapel, im Augenblick gerade nicht verarbeitete Adressen werden während eines Unterprogrammes oder einer Unterbrechung gespeichert, um danach sofort wieder zur Verfügung zu stehen.

Akkumulator: Rechenwerkregister, in dem das Ergebnis arithmetischer Operationen gebildet wird. Auch zur Zwischenspeicherung benutzt.

Algol: Algorithmic **L**anguage Problemorientierte Programmiersprache für technisch-wissenschaftliche Programme

alphanumerisch: aus Ziffern und Buchstaben gemischte Darstellung

ALU: Arithmetic **L**ogic **U**nit, arithmetische und logische Einheit, Rechenwerk

Arbeitsgeschwindigkeit: Sie ergibt sich aus der Technologie und der Organisation des Mikrocomputersystems. Während Mikroprozessoren der ersten Generation für die Addition eines Datenwortes etwa 20 µs benötigen, führt die zweite Generation in

MOS-Technik diese in 2 μs aus. Bei bipolaren Ausführungen sind Größenordnungen von 100 ns erreichbar.

Arbeitsspeicher: In der Regel ein (Kurzzeit)-Zwischenspeicher, in großen EDV-Anlagen auch Zentralspeicher

ASCII: **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange. Amerikanische Kode für Informationsaustausch

Assembler: sowohl maschinenorientierte Programmsprache als auch Übersetzungsprogramm zur Umwandlung dieser Sprache in die binäre Maschinensprache

Asterik: Stern (Zeichen)

Asynchron: Taktunabhängig

Auxiliary function: Hilfsfunktion

Background program: Hauptprogramm

BASIC: **B**eginners **a**ll purpose **s**ymbolic **i**nstruction **c**ode Dialogorientierte Programmiersprache

Basic instruction: Grundbefehl

BCD-Code: **B**inary **C**oded **D**ecimal-Code, Binärkode für Dezimalzahlen

Bedingter Befehl: Befehl, dessen Ausführung vom Inhalt bestimmter Speicherplätze verschieden sein kann. (z.B. nach einer Entscheidung)

Bedingte Kennzeichen: Englisch: conditional flags

Bedinger Sprungbefehl: Sprungbefehl, der nur befolgt wird, wenn eine bestimmte Voraussetzung (z.B. Ergebnis eines Vergleiches) vorliegt

Befehl: Anweisung an den Computer, die sich in der benutzten Sprache nicht mehr in Teile zerlegen läßt, die selbst Anweisungen sind.

Befehlsdiagramm: Detaillierte Darstellung der Befehlsfolge, geeignet zum Kodieren des Programmes

Befehlsfolgeregister: auch Befehlszähler genannt, gibt der CPU an, welches der jeweils nächste Befehl ist, der in das Steuerwerk eingelesen werden muß.

Befehlsvorrat: Summe der zulässigen Befehle, die ein gewisser Computertyp ausführen kann

Befehlszähler: Programmzähler

Benchmark program: Bewertungsprogramm

Betriebssystem: Reihe von Programmen, auch als Systemprogramme bezeichnet, die zum Betreiben eines Computers benötigt werden

Bidirektional: zweigleisig, Zweiweg, Signale können in beiden Richtungen übertragen werden (Zeitmultiplex)

Bit: kleinste Darstellungseinheit für Binärdaten

Boolesche Algebra: Rechnen mit dualem Zahlensystem (Mathematiker G. Boole 1840)

Boolesche Funktion: Beziehung zwischen mehreren (binären) Variablen

Bootstrap-loader: Urlader

Borrow: negativer Übertrag

Branch: Verzweigung

Branch address: Verzweigungsadresse

Breakpoint: Unterbrechungstabelle

Buffer: Puffer, Speicher in dem Daten zwischengespeichert werden

Bug: Störung, zeitweises Versagen, Fehler im Programm

Bus: Sammelschiene, an der mehrere Bausteine angeschlossen sind

BUSEN: **Bus en**able, Sammelschienen-Freigabe

Byte: Speicherstelle mit 8 Bits

CAD: **C**omputer **a**ided **d**esign, rechnergestützte Entwicklung

Calculation: Berechnung

CALL: Aufruf, z.B. eines Unterprogrammes

Call-in: Aufruf

Call instruction: Rufbefehl

Carry: Übertrag

Cartridge: Magnetbandkassette

Character: (Schrift-)Zeichen, Symbol

to check: kontrollieren, prüfen

Chip: monolithischer Halbleiterkristall

Chip-Enable: Freigabe eines Bausteines über einen Freigabe-Eingang

Chip-Select: Baustein-Auswahl (z.B. bei mehreren Speicherbausteinen)

to clear: Löschen, freigeben

Clock: Takt, Zeitgeber, Taktimpuls

Clock generator: Taktgeber

CMA: **C**o**m**plement **A**ccumulator, der Inhalt des Akkumulators soll komplementiert werden

CMC: **C**o**m**plement **C**arry, der Übertrag soll komplementiert werden

Cobol: **Co**mmon **B**usines **O**riented **L**anguage, Programmiersprache für kaufmännische Aufgaben

Code: Verschlüsselung, Vorschrift für die eindeutige Zuordnung der Zeichen eines Zeichenvorrates zu denjenigen eines anderen Zeichenvorrates (Bildmenge).

Codieren: In der Datenverarbeitung: Ein Programm aus einem als Diagramm festgehaltenen Arbeitsablauf (Programmablaufplan) schreiben

to collate: ordnen, zuordnen, mischen

Comment: Kommentar

C-MOS: **C**omplementary **MOS**, MOS-Technologie mit P- und N-Kanal-Transistoren

Compiler: Übersetzer (einer Programmiersprache in ein Maschinenprogramm)

Complement: Komplement, Ergänzung

Computer: Rechengerät, Rechenanlage, EDV-Anlage, Logik, die programmierbar ist

Conditional: bedingt

Conditional branching: bedingte Verzweigung

Conditional flags: bedingte Kennzeichen

Control bus: Steuer-Sammelschiene

Control Unit: Steuerwerk, Kommandowerk, Leitwerk

Control word: Steuerwort

Conversion: Umkodierung, Übersetzung

Core: (Magnet-)Kern (speicher)

CPU: **C**entral **P**rocessing **U**nit, ein Teil der Zentraleinheit, häufig wird auch der Mikroprozessor so bezeichnet

Cross-Assembler: Programmübersetzer

Cross-Compiler: Programmübersetzer von einem in ein anderes Programm

CRT: **C**athode **R**ay **T**ube, Kathodenstrahlröhre, Bildschirmterminal

Cycle: Zyklus, Schleife

DAA: **D**ecimal **A**rithmetic **A**djustment, das Rechenwerk kann dabei Dezimalzahlen verarbeiten

Data-Bus: Datenbus, Datensammelschiene

Date processing: Datenverarbeitung, jedes Verarbeiten digitaler oder analoger Daten (Informationen). Die wichtigsten Tätigkeiten in der DV sind: Reproduzieren, Identifizieren, Vergleichen, Sortieren, Zuordnen, Komprimieren und Rechnen

Date mode signal: Datenart-Kennung

Daten: Numerische oder alphanumerische Angaben über Sachverhalte und Dinge

Datenbus: Eine Sammelschiene, die den Mirkoprozessor mit den Speicherbausteinen sowie den Eingabe- und Ausgabeeinheiten verbindet. Intern wird auch ein großer Teil der Ablaufsteuerungen über diese Schiene geleitet. Die Sammelschiene kann in beide Richtungen Daten übermitteln. Die angeschlossenen Bausteine können drei "Zustände" an ihren Klemmen haben (Tristate): Low, High und hochohmig. Mit den Pegeln "L" und "H" werden Signale an die Sammelschiene abgegeben. Der Zustand "hochohmig" dient zum Empfang von Signalen aus der Sammelschiene.

Datenflußplan: Darstellung der zeitlichen Folge der Einzelprogramme

Datensammelschiene: Verbindungen, bei denen (nach dem Multiplexverfahren) meist in beiden Richtungen eine Vielzahl von Informationen übertragen wird, siehe auch "Datenbus".

Datenspeicher: Speicher, in denen ausschließlich Daten gespeichert sind, meist RAM

DA-Wandler: **D**igital-**A**nalog-Wandler

DBIN: **D**ata **B**us **In**put, Datensammelschiene ist zur Aufnahme von Informationen bereit

Debugging: Fehlerbeseitigung

Decimal adjust: Dezimalkorrektur

Decrement: Dekrement, schrittweises Erniedrigen um -1

Dedicated: spezifisch, anwendungsspezifisch

Dekodieren: Entschlüsseln

Density: Dichte

Destructive: löschend

Development system: Entwicklungssystem

Device: Einheit, Gerät

D-Flip-Flop: D-FF, Zwischenspeicher, der so gesteuert wird, daß die Information erst beim nachfolgenden Taktimpuls weitergeht

Dialoggerät: Gerät zur unmittelbaren Befehls- und Dateneingabe sowie Datenausgabe

Digit: Ziffer, Stelle

Digitaltechnik: In der Digitaltechnik kennt man nur zwei Zustände, nämlich "ein" und "aus". Die zugehörigen Pegelwerte werden nach DIN-Norm 41785 Blatt 4 mit "L" (Low) und "H" (High) bezeichnet

Direct access: Direkter Zugriff

Disk storage: Plattenspeicher

Display: Anzeige

DMA: **D**irect **M**emory **A**ccess, direkter Speicherzugriff, ohne daß die Information über die CPU gehen muß

Down time: Totzeit, Ausfallzeit

Drain: Senke, Elektrode bei Feldeffekttransistoren, die dem Kollektor bei bipolaren Transistoren entspricht

Drum storage: Trommelspeicher

to dump: abschalten, (Fehler) ausgeben

Durchsatz: Anzahl von Operationen, die ein Computer pro Zeiteinheit leistet

Dynamischer Speicher: Speicher, bei dem die Information zyklisch aufgefrischt werden muß

EAROM: **E**lectrical **A**lterable **R**ead **O**nly **M**emory Festwertspeicher, meist in MNOS-Technik, dessen Inhalt sich elektrisch verändern läßt

EBCDI-Code: **E**xtended **B**inary **C**oded **D**ecimal **I**nterchange-Code

ECL: **E**mitter **C**oupled **L**ogic, besonders schnell arbeitende Logikfamilie, bei der die Transistoren nicht in die Sättigung gesteuert werden.

Editor: Programmierhilfe

EDV: **E**lektronische **D**aten-**V**erarbeitung

Ein-Adreß-Befehl: Befehl mit nur einer Speicheradresse

Ein-Wort-Befehl: Befehl, welcher nicht mehr Bit als die Wortlänge des Computers umfaßt (z.B. 8 oder 16 Bit)

Emulator: Programm, das die Eigenschaften eines kleineren Computers auf einem größeren Computer nachbildet

Enable (-Signal): Freigabe (-Signal)

Encoding: Kodierung

EPROM: **E**rasable-**P**rogrammable **R**ead **O**nly **M**emory (Mit ultraviolettem Licht) löschbarer programmierbarer Festwertspeicher

to erase: löschen

Even: geradzahlig

Exess-three-code: Dreiexeß-Code

Execution time: Zeit zur Ausführung eines Befehls, Ausführungszeit

Exorciser: Ges. gesch. Ausdruck (Motorola) für ein Hilfsgerät zur Entwicklung von Mikrocomputer-Systemen

External acknowledge: Externe Quittung

External storage (device): Externer Speicher, außerhalb des Computer sich befindendes Speichermedium, z.B. Karten- oder Bandgeräte

FAMOS: **F**loating **A**valanche **I**njection **MOS,** Technologie, die bei Reproms angewendet wird

Fan in: Eingangslastfaktor

Fan out: Ausgangslastfaktor, Anzahl von Bausteinen gleicher Logikserie, die an einen Aus-

gang angeschlossen werden können.

to feed: speisen

Fetch machine cycle: Abrufzyklus

Festwertspeicher: Englisch: ROM, **R**ead **O**nly **M**emory, auch Festspeicher oder Lesespeicher, man kann nur *feste* Daten auslesen

FF: Flip-Flop, elektronischer Speicher

Field: Feld, Gruppe

Fifo: **F**irst **i**n **f**irst **o**ut, zuerst eingehende Daten werden auch zuerst ausgelesen.

Figure: Ziffer

File: Datenmenge

First state: Anfangszustand

Fixed-point: Festkomma

Flag: Kennzeichen, werden z.B. bei bedingten Verzweigungen gesetzt

Flag register: Kennzeichenregister

Floating gate: nicht fest angeschlossene Steuerelektrode bei Feldeffekttransistoren, der von außen eine Ladung aufgeprägt werden kann. (Wird bei PROMs verwendet)

Floating output: potentialfreier Ausgang

Flouting-point: Gleitkomma

Floppy Disk: billiges Plattenspeichermedium, entwickelt für Mikrocomputer

Flow chart: Flußdiagramm, Befehlsschema

Flüchtige Speicher: Speicher, deren Inhalt bei Stromausfall verloren geht

FORTRAN: **For**mula **Tran**slation problemorientierte Programmiersprache für technisch/wissenschaftliche Programme

FPLA: **F**ield **P**rogrammable **L**ogic **A**rrays, logische Matrix, bei der vom Anwender bestimmt wird, ob sie als ODER bzw. UND-Schaltung arbeitet.

Front panel control: Bedienungsfeld-Steuerung

Funktions-Einheit: Zu einer Einheit zusammengefaßte Schaltungen, Bausteine oder Baugruppen, die im wesentlichen zur Ausführung einer Funktion dienen

Gap: Lücke, Spalt

Gate: Steuerelektrode bei Feldeffekttransistoren

Gate circuit: (Abgekürzt: Gate) Gatter, Torschaltung

Gating array: Gatter-Matrix

Handler: Automatische Prüfeinrichtung

to halt: stoppen, unterbrechen

HALT: Halt-Befehl, wird durch Programm eingegeben

Handshaking: Quittungsbetrieb, Austausch von Synchronisationsimpulsen für Geräte mit verschiedenen Reaktionsgeschwindigkeiten

Hardware: Baustein, technische Ausrüstung, Anlage

Hardwired Logic: Logische Schaltung aus Standardbausteinen zusammengefügt

High: Logikpegel nach DIN 41785 Blatt. Bei positiver Speisespannung entspricht "H" dem Zustand "ein" (Logische eins).

HLTA: HALT-Acknowledge, Quittung für Halt-Befehle

HOLD: Belegt halten, Befehl wird durch Steuerleitung eingegeben, bewirkt, daß Daten-

sammelschiene hochohmigen Zustand annimmt und für externe Zwecke (z.B. DMA) belegt gehalten wird.

Hopper: Kartenmagazin

Hub (engl.): Buchse (Schalttafel)

ICE: Emulations- und Test-Adapter

Immediate Data: Direktdaten

Increment: Inkrement, schrittweises Erhöhen um 1, z.B. der Programmzähler zählt um einen Schritt weiter.

Indexregister: werden zur Änderung von in Befehlen enthaltenen Adressen, für Zähloperation und Einleitung von Verzweigungen verwendet, sie sind zur Zwischenspeicherung der Basisadressen bestimmt.

Inhibit: Sperr........

INP: **Inp**ut, Daten-Eingabe

Input port: Eingabe-Baustein

Input signal: Eingangs-Signal

Instruction: Befehl

Instruction decoder: Befehlsdekoder

Instruction set: Befehlssatz, Befehlsliste

INT: **Int**errupt request, Unterbrechungsanfrage vom Eingang her

INTA: **Int**errupt-**a**cknowledge, Quittung für Unterbrechungsgesuch

INTE: **Int**errupt **e**nable, Unterbrechung, gesteuert von der CPU für Ausgabe

Integer: ganze Zahl

Interface: nach beiden Seiten angepaßte Trennstelle, Schnittstelle

Interface control: Anpassung

Interface-Logik: Logik-Schaltung zur Anpassung

Interpreter: Zuordner, Zuordnungsprogramm

INTERRUPT: Unterbrechung, Unterbrechungsbefehl

Interrupt request: Unterbrechungsanfrage von einer externen Einheit, um z.B. Daten einzugeben

Interrupt request latch: Unterbrechungsanforderungs-Flip-Flop

Interrupt state acknowledge signal: Unterbrechungs-Quittiersignal

Invalid: ungültig

I/O: Ein-Ausgabe

I/O-instruction: Eingabe-Ausgabe-Befehl

Ionenimplantation: Dotierungsverfahren anstelle von Diffusion. Erreicht geringere Fertigungstoleranzen, damit bei MOS-Technik auch niedrigere Schwellspannungen (siehe Depletion-load)

IS: Integrierte Schaltung, in der Weltliteratur als IC (**I**ntegrated **C**ircuit) bezeichnet

I^{2}L: **I**ntegrated **I**njection **L**ogic, neue bipolare Technologie mit sehr hoher Packungsdichte

JUMP: Sprung, z.B. an eine andere Programmstelle

JUMP CONDITIONAL: bedingter Sprung

JUMP, -COMPUTED: errechneter Sprung

kByte: Kilo-Byte = 2^{10} = 1024 Byte

Kernspeicher: Speicher aus Ferritkernen

Keyboard: Tastatur

Kompatibel: austauschbar, aneinander angepaßt

Label: Merkmal, Kurzzeichen, Kennsatz, Wortsymbol

Last in/first out: zuletzt eingeschrieben wird zuerst ausgelesen

Latch: Elektronischer Schalter mit Halteanordnung (z.Teil als D-Flip-Flop ausgeführt), Signalspeicher, Auffangspeicher

LED: **L**ight **E**mitting **D**iode, Lumineszensdiode, gebräuchlich in den Farben Gelb, Grün und Rot

Leitwerk: Steuerwerk, Kommandowerk

Library: Bibliothek

Lifo: **L**ast **I**n **F**irst **O**ut, zuletzt gespeicherte Daten werden zuerst ausgegeben

Line: (Strom-)Netz

LMI: **L**oad **M**emory **I**mmediate (with the next Byte in memory following the instruction byte) Einschreibebefehl

LOAD: laden (Programm), auch umwandeln

Loader: Ladeprogramm

Logic: Logik, (logische) Digitalschaltung

Loop: Schleife

Lowest-level-register: Register, dessen Inhalt zu allererst eingeschrieben wurde. Bei mangelnder Registerkapazität wird der Inhalt gelöscht, um neue Informationen stapeln zu können

Low order position: niederwertige Stelle

Low-power-TTL: TTL-IS mit geringerem Leistungsbedarf, zunächst langsamer arbeitend. Durch Einführung von Schottky-Dioden der Standard-TTL hinsichtlich Arbeitsgeschwindigkeit gleichwertig.

LSI: **L**arge **S**cale **I**ntegration, großer Integrationsumfang

Machine cycle: Maschinen-Zyklus, Operationszyklus

MACRO: Befehle durch Assembler

MACRO-Assembler: Ein Übersetzer vom Ursprungsprogramm in Zielanweisungen für die Maschine

Magnetkarten: Billiges externes Speichermedium, z.B. in programmierbaren Taschenrechnern

Makro: Zusammenfassung von Befehlen

Marginal check: Grenzwertprüfung

Maschinencode: Maschinensprache, Binär-Code, die vom Computer direkt verstanden wird

Maskenprogrammierung: Festlegung des Speicherinhaltes eines ROM durch die Verdrahtungsmaske

Mass storage: Großspeicher

to match: anpassen, vergleichen

MCS: **M**icro **C**omputer **S**ystem

MDS: **M**icrocomputer **d**evelopment system, Entwicklungssystem für Mikrocomputer

Medium scale integration: mittlerer Integrationsumfang

Mehradreßbefehl: Befehl, bei dem mehrere Adressen angesprochen werden.

Mehrwortbefehl: Befehl mit einer größeren Wortlänge als die des Computers. Muß entsprechend seiner Länge

in mehreren Speicherplätzen untergebracht werden.

Memory: Speicher

MEMR: Memory **r**ead, d.h. aus dem Speicher soll gelesen werden

to merge: (zusammen)mischen

Microinstruction: Mikrobefehl, Teil eines Mikroprogramms

Mikrocomputer: Computer mit einem Mikroprozessor. Anwendungsgebiete von Mikrocomputern: Ersatz von nicht ausgelasteten Minicomputern und festverdrahteter Logik. Die Wirtschaftlichkeitsgrenze liegt zwischen 30 bis 90 Logikbausteinen (MSI) und hängt weitgehend von den auszuführenden Operationen ab. Einsatz z.B. bei der Steuerung von Prozessen und Maschinen, für die Überwachung von Anlagen und Geräten, bei der Vermittlung von Nachrichten- bzw. Datenverbindungen

Microprogrammable: mikroprogrammierbar

Microprocessor: Microprogrammable central **processing** unit

Mikroprogramm: Ablaufsteuerung für eine Zentraleinheit, stromausfallsicher gespeichert, in einem ROM. Bei den schnellen, aus Prozessorelementen zusammengesetzten Mikroprozessoren muß der Anwender das Mikroprogramm in binärem Code selbst erstellen. Summe der Mikroprogramme realisiert Maschinenbefehlsvorrat. Die meisten Mikroprozessoren sind nicht mikroprogrammierbar, sondern haben einen festen Befehlsvorrat.

Mikroprogrammierbarkeit: Eigenschaft einer Ablaufsteuerung in einem Steuerwerk. Die Befehle werden aus Mikrobefehlen zusammengesetzt, können selbst erstellt werden und sind in einem Festwertspeicher aufbewahrt. Gegenteil: Festverdrahtete Befehle

Mikroprozessor: Integrierter Standardbaustein als Teil eines Mikrocomputers, enthält Rechen- und Leitwerk, deren interner Ablauf in der Regel von außen mit einem (programmierbaren) Festwertspeicher beeinflußbar ist. Der Mikroprozessor besteht entweder aus einem Chip oder wird bei den schnelleren bipolaren Typen aus mehreren Chips, den sogenannten Prozessorelementen (Slices) zusammengesetzt.

Mikroprozessorpaket: Das Mikroprozessorpaket ist ein Bausteinsatz, bestehend aus Mikroprozessor, Speicherbausteinen (RAM, ROM, PROM), Eingabe- und Ausgabe-Auffangspeichern, Taktgenerator und Hilfsstufen (Systemsteuerung, Daten-Interface, Decoder, Pegelanpassung) zum Aufbau kompletter Mikrocomputer, die außer in zusammengesetzten Ausführungen (mehrere Bausteine) auch als Ein-Chip-Ausführung bekannt sind.

Mnemonischer Code: Alphanumerische Kürzung für Befehle, sind leichter zu merken als Binärcodes.

MNOS: Metall **N**itrod **O**xid **S**emiconductor. Technologie zum Bau von EAROMs

Modem: Modulation and **Dem**odulation **Modul**, Schaltung zur Fernübertragung von Daten (z.B. über Fernsprechleitungen)

Module (engl.): Modul, Einheit

Monitor: Programmsteuersystem, kann auch aus "Software" bestehen

MOS: Metall-**O**xyd-**S**emiconductor, Metall-Oxyd-Halbleiter

MOVE: verschieben (Daten)

MP: Mikro**p**rozessor

MPU: Micro **p**rocessing **u**nit, Mikroprozessor

MSI: Medium **S**cale **I**ntegration, mittlerer Integrationsumfang

Multiaddress instruction: Mehradreßbefehl

Multilevel interrupt: mehrstufige Programmunterbrechung

Multiplex: Mehrfachübertragung, meist zeitlich hintereinander folgende Informationen (Zeitmultiplex)

Multiprocessing: Problembearbeitung durch mehrere Rechenwerke oder Computer

Multiprogramming: Programmverzahnung, Ausführung mehrerer Programme durch einen Computer

Multisystem: Zusammenschluß mehrerer Systeme

M-Zyklus: Maschinen-Zyklus, Operationszyklus, englisch: cycle

M 1: Maschinen-Zyklus 1

NC-Maschinensteuerung: Numeric **C**ontrol-Maschinensteuerung, Steuerung einer Maschine über Lochstreifen

Nesting: Einschachtelung, Verschachtelung

N-Kanal-MOS-Technologie: auf P-Substrat aufgebaut MOS-Transistoren, bei denen die gesteuerten Zonen (Kanal) N-dotiert sind

Von Neumann-Zyklus: Zusammenfassung aller Maschinen-Zyklen

Non-volatile: nicht flüchtig

Notation: Schreibweise, Darstellung

Number: Zahl

Odd: ungerad(zahlig)

Off-line: Von der Anlage getrennt, Verkehr wird über Datenträger abgewickelt, unabhängiger Betrieb

On-line: Mit der Anlage verbunden, schritthaltend miteinander synchronisiert, abhängiger Betrieb

Operand: Rechengröße

Operation: sowohl Befehl (kleinster Teil eines Programmes) als auch Operation, wie Addieren, Subtrahieren etc.

Operationsregister: Befehlsfolgeregister

Operator: Bedienungsperson (für Computer)

OUT: **Out**put, Daten-Ausgabe

Output buffer: Ausgabepuffer

Overflow: Überlauf

PACE: Processing **a**nd **c**ontrol **e**lement (Mikroprozessor)

Packen: Zwei Dezimalzahlen in einem Byte unterbringen

Packing density: Aufzeichnungsdichte

Packungsdichte: Anzahl der Einzelelemente, (meist Transistoren, Dioden, Widerstände u. Kondensatoren) auf einem Chip

Parity: Parität, Gleichheit

Parity bit: Paritätsbit

Parity check: Vollständigkeits-, Paritätskontrolle

Patchboard: Schalttafel

Pattern: Muster, Figur, Zeichen

PCC: **P**ut **c**ontrol **c**all, Input-Output cycle, Ein-Ausgabe-Zyklus

PCI: **P**ut **c**ontrol **i**nstruction, Instruction fetch cycle, Befehlsabruf-Zyklus

PCR: **P**ut **c**ontrol **r**ead, Memory read cycle, Speicherlese-Zyklus

PCW: **P**ut **c**ontrol **w**rite, Memory write cycle, Speicher-Einschreibe-Zyklus

Periphere Einheiten: Sammelbegriff aller technischen Aggregate, die an eine Zentraleinheit angeschlossen werden können

Peripheral: Peripherieeinheit

Periphere Geräte: Externe Geräte, z.B. Floppy-Disk, Terminals, Sichtgeräte

PIA: **P**eripheral **I**nterface **A**dapter, μC-Baustein für Ein- und Ausgabe

Pin: Anschlußstift

Pin connection: Anschlußbelegung

Pinkompatibel: Sämtliche Anschlüsse sind gleich, so daß die integrierte Schaltung direkt austauschbar ist, dabei braucht die Innenschaltung nicht übereinzustimmen.

PLA: **P**rogrammable **L**ogic **A**rray, programmierbare Logikanordnung, bestehend aus ROM's. Matrix in drei Komplexen:
1. Produkttermmatrix (UND-Gatter)
2. "ODER"-Matrix
3. Programmierteil der Ausgänge

PL/M: **P**rogramming **L**anguage for **M**icrocomputers Programmiersprache für Mikrocomputer

Plotter: (programmgesteuerter) Kurvenschreiber

P-Kanal-MOS-Technik: auf N-Substrat aufgebaute MOS-Transistoren bei denen die gesteuerte Zone (Kanal) P-dotiert ist.

Point: Komma (bei Zahlen)

Pointer: Zeiger

Polling: zyklische Abfrage

Port: Kanal, Ein- oder Ausgabe-Baustein

Power: Potenz (einer Zahl)

to preset: vorgeben

Primärprogramm: Meist in einer problemorientierten Sprache geschriebenes Programm, welches mittels Compiler in ein Maschinenprogramm umgesetzt wird

Priority: Priorität

Priority Encoder: Baustein oder Schaltung, die die Priorität angeschlossener Geräte bei der Eingabe festlegt

to process: verarbeiten

Processor: Zentraleinheit

Programm: Summe aller Befehle zur Ausführung einer Operation

Programmablauf: Darstellung aller beim Programmablauf möglicher Wege. Man unterscheidet Datenflußplan, Programmablaufplan und Befehlsdiagramm. Der Programmablaufplan nimmt zwischen dem Datenflußplan und dem Befehlsdiagramm eine Mittelstellung ein.

Programmierbare Festwertspeicher: Englisch: PROM, **P**rogrammable **R**ead **O**nly **M**emory, nachträglich programmierbarer Festwertspeicher

Programmierung: Ein Programm umfaßt die Summe aller Befehle, die der Computer braucht, um eine Verarbeitung, z.B. eine Rechnung, eine Sortierung oder eine andere Aufgabe ausführen zu können. Es dürfen nur Befehle genommen werden, die im Befehlsvorrat des Mikroprozessors enthalten sind. Der Mikroprozessor versteht nur seine eigene Maschinensprache. Von großem Vorteil ist, daß für die verschiedensten Aufgaben nur ein außen angeschalteter programmierbarer Speicher (ROM bzw. PROM) ausgewechselt zu werden braucht.

Program Counter: Programmzähler, Befehlszähler

Programmiersprachen: Sprache zur Formulierung der Programme, bestehend aus festen Befehlssätzen, die mit einem Compiler in die Maschinensprache übersetzt werden können.

Programmiersprachen, problemorientierte: Sie werden auch höhere Programmsprachen genannt und haben einen engeren Bezug zur wissenschaftlichen Darstellung. Gegenüber Maschinensprachen sind die Programme im allgemeinen übersichtlicher und einfacher zu erstellen. Für Mikrocomputer gibt es z.B. die Programmiersprache PL/M, die mit einem Compiler in die Maschinensprache umsetzbar ist.

Programmzähler: Befehlszähler

PROM: **P**rogrammable **R**ead **O**nly **M**emory, programmierbarer Lesespeicher

PROM-Programmiereinrichtung: Einrichtung zur vollautomatischen Programmierung von PROMs

Pufferspeicher: Zwischenspeicher

to punch: stanzen, lochen

Punch card: Lochkarte

Punched tape: Lochstreifen

PUSH: Schieben, z.B. bedeutet "PUSH B", daß angegebener Inhalt B gespeichert wird

Quittungsbetrieb: Datenverkehr mit Austausch von Synchronisationsimpulsen für Geräte mit verschiedenen Reaktionsgeschwindigkeiten

RALU: **R**egister and **A**rithmetic **L**ogic **U**nit. Prozessorelement mit ALU und einigen Registern

RAM: **R**andom **A**ccess **Me**mory, Speicher mit direktem Zugriff, z.B. Schreib-/Lesespeicher

Random: beliebig, Zufalls......

Random access: Wahlfreier Zugriff, jede Speicherstelle kann direkt angesprochen werden

READY: betriebsbereit, z.B. Speicher ist zur Abgabe von Daten bereit

Ready signal: "bereit"-Signal

Real-Time: Betrifft das unmittelbare Zeitintervall, in dem ein physikaler Prozess sich ereignet, Echtzeit

Reader: (Lochstreifen- oder Lochkarten-) Leser

Rechenwerk: Teil einer EDV, siehe Abschnitt 2

to record: aufzeichnen

Record: Block, Satz

Redundanz: Die Mitübertragung von Informationen, die zur Datensicherung dienen und in störungsfreiem Betrieb ohne Informationsverlust fortgelassen werden können.

Refresh: Wiederauffrischung, Auffrischen, wird bei dynamischen Speichern benötigt, damit ihr Inhalt nicht verlorengeht, Bildwiederholung (bei CRT)

Register: Kurzzeitspeicher, Speicher

Register Select: Register-Auswahl

Reliability: Zuverlässigkeit

REPROM: **Rep**rogrammable **r**ead **o**nly **m**emory. Festwertspeicher, der sich löschen und wieder neu programmieren läßt

Request: Anfordern, Anforderung

to rerun: wiederholen

to reset: (zu)rücksetzen, löschen

RESTART: Wiederanlauf nach einer Unterbrechung

to restore: wiederherstellen

RETURN: Rückkehr, z.B. an die Programmstelle, ab der ein Unterprogramm eingefügt wurde

to rewind: rückspulen

ROM: **R**ead **O**nly **M**emory, Lesespeicher mit festem Inhalt

to round off: (auf-)runden

Routine: Programm, auch Unterprogramm

RST: **rest**art, Befehl zum Wiederanlauf nach einer Unterbrechung

Run: Durchlauf

SCAMP: **S**imple to use **c**ost effective **a**pplication **m**icro**p**rocessing, Firmenbezeichnung für einen Mikroprozessor

to scan: abtasten

Schreib-/Lesespeicher: Englisch: RAM, Random Access Memory, Speicher, dessen Inhalt sich überschreiben läßt

Scratch pad: Arbeitspuffer

Scratch pad memory: schneller Hilfsspeicher, "Notizbuchspeicher"

Schnittstelle: Pegel- und Anschlußgenormte Trennstelle zwischen zwei Geräten

Second source: Möglichkeit, ein identisches Bauelement von einem zweiten oder mehreren anderen Herstellern zu beziehen

Select: Auswahl

to sense: abtasten

Serial: seriell

Set: Gerät, Satz

to set: setzen

Shift: Verschiebung

Shift-register: Schieberegister

Sign (Bit): Vorzeichen

Simulator: Befehlsnachbildungsprogramm, meist zum Testen eines erstellten Programmes
Skew: Schräglauf
to skip: springen, vorschieben
Slice: (Prozessor-)Element
Small scale integration: Kleiner Integrationsumfang
Software: Programm, Summe aller Befehle
to sort: sortieren
SOS: Silicon **O**n **S**apphire, MOS-Technologie
Source: Quelle, Elektrode bei Feldeffekttransistoren, die dem Emitter bei bipolaren Transistoren entspricht
Space: (Zwischen-)raum
Speicher: Medium oder Schaltungsanordnung, die Daten festhalten können. Man unterscheidet zwischen flüchtigen und nicht flüchtigen Speichern. Erstere verlieren bei Stromausfall ihren Inhalt, letztere behalten ihn bei. In einem Mikrocomputersystem werden Festwertspeicher (ROM, PROM und REPROM) zur Speicherung des Programms, Schreib-Lese-Speicher (RAM) zur Aufnahme der Daten eingesetzt. Ist man sich bei der Entwicklung des Programms noch nicht sicher, ob es alle gewünschten Funktionen ausführt, kann bei verschiedenen Mikroprozessoren über ein Ladeprogramm der Maschinenkode-Lochstreifen in einen RAM-Speicher geladen werden. Nach dem Erproben des Programms kann entweder beim Halbleiterhersteller das passende ROM bestellt werden, oder der Anwender programmiert sich ein PROM selbst.
Speicherorganisation: Zuordnung der Speicherzellen zu bestimmten Adressen
SSI: Small **S**cale **I**ntegration, kleiner Integrationsumfang
Stack: Stapelspeicher, Kellerspeicher, Stapelregister, Speicherstellen, die zur Aufbewahrung von Registerinhalten innerhalb oder außerhalb des Datenspeichers vorgesehen sind.
Stack-Pointer: Stapelzeiger, Register in dem die (Stack-) Speicheradresse gespeichert ist.
Standardbausteine: Frei verkäufliche Bausteine, Gegensatz zu kundenspezifischen Bausteinen
State: Zustand, Operationsschritt, T-Zyklus (kleinste Zeiteinheit des Operationszyklus (M-Zyklus))
Statement: Anweisung, Befehl, eine in einer beliebigen Sprache abgefaßte Arbeitsvorschrift. Wichtige Klassen sind:
Arithmetische Anweisung,
Boolesche Anweisung,
Verzweigungsanweisung,
Sprunganweisung,
Transportanweisung
Statischer Speicher: Speicherbaustein, der keinen Wiederauffrischungszyklus benötigt
Status: Zustand
Status latch: Zustandssignal-Speicher

Status-strobe: Impuls zur "Momentaufnahme" eines Zeitablaufs
STC: **S**et **C**arry, Übertrag-Bit wird direkt auf "H" gesetzt
Steuerwerk: auch Leitwerk genannt, Teil einer EDV (siehe Abschnitt 2) Englisch: Control unit
STOPPED: Angehalten
Storage: Speicher, Speicherung
STORE: Speichern
STSTB: Statusabfrage
Subroutine: Unterprogramm
Substrat: Halbleiterscheibe, auf der durch Diffusionen und/oder Ionenimplantation integrierte Schaltungen bzw. Transistoren hergestellt werden
Supervisor: Organisationsprogramm
System: Komplexe Geräteanordnung
Systemanalyse: Bearbeitungsphase in der Analyse der Struktur eines Problems
Tag (engl.): Marke
Takt: Periodisches Signal zur Synchronisation
Taktgeber: Pulsgenerator zur Synchronisierung von Operationen in digitalen Rechenanlagen
Tape: (Magnet-)Band, (Loch-) Streifen
Temporary register: Zwischenspeicher-Register
Terminal: Anschlußeinheit, z.B. an Computer
Text-Editor: Programm zur Erstellung eines immer wiederkehrenden Textes
Throughput: Durchflußleistung, Datendurchlauf, Maß für die System-Leistungsfähigkeit, Durchsatz
Time sharing: Zeitlich verzahnte Verarbeitung mehrerer Programme
Timing: Zeitl. Zuordnung, zeitl. Verlauf, zeitl. Steuerung
Track: Spur, Bahn
Transfer: Übertragung, übertragen
Transfer request: Übertragungs-Anforderung
Tristate: siehe Datenbus
Tri-stata-gate: Gatter mit drei Ausgangszuständen ("H", "L" und hochohmig)
to truncate: runden
Truth table: Wahrheitstabelle
Trunk: Kanal, Weg Sammelschiene
TTL: **T**ransistor **T**ransistor **L**ogic, digitale Standardbausteinreihe mit 5 V Versorgungsspannung
TTL-kompatibel: zum Anschluß an TTL-Schaltungen geeignet
TTY: Teletype, Fernschreiber, häufig mit Lochstreifenleser und Stanzer
T-Zyklus: Englisch: "State", meist einer Taktperiode entsprechend
Unconditional: unbedingt
Unit: Gerät, Einheit
Unterprogramm: Englisch: Subroutine, eines der wichtigsten Hilfsmittel zur Rationalisierung der Programmerstellung. Gleiche Operationen, die an verschiedenen Stellen eines Programmes benötigt werden, brauchen als Unterprogramm nur einmalig erstellt zu werden. Mikroprozessoren kön-

nen ähnlich wie große EDV-Anlagen Unterprogramme verarbeiten. Während der Ausführung des Unterprogramms wird die Adresse des letzten Befehls in einem sogenannten "Stapel" aufbewahrt (zwischengespeichert). Dieser Zwischenspeicher ist entweder im Mikroprozessor selbst untergebracht, oder es werden mit Hilfe eines Zeigers ("stack-pointer") Speicherplätze im RAM angesprochen.

Valid: gültig

Vektor Interrupt: Unterbrechung, wobei jedes anfordernde Gerät seine eigene Service-Routine hat oder bekommt

Vektoroperation: Zeigerbefehl

VLSI: **V**ery **L**arge **S**cale **I**ntegration (z.B. Mikrocomputer auf einem Chip)

Volatile: Flüchtig

Wahlfreier Zugriff: Direktzugriffsverfahren, jede Speicherstelle kann direkt angesprochen werden, unabhängig von der Aufzeichnungsreihenfolge

WAIT: warten

Wait state: Wartezustand

Waveform: Impulsform

Wiederauffrischung: Englisch: refresh, wird bei dynamischen Speichern benötigt, damit ihr Inhalt nicht verloren geht

$\overline{\text{WO}}$: Negation von **W**rite **o**utput, Speichereinschreibetätigkeit oder Ausgabe von Daten

Worst case: ungünstigster Fall

Worst case pattern: kritisches Muster

Wort: Zusammenfassung mehrerer Bits oder Speicherstellen. Je nach Anlage verschieden (10-64 Bits), Datenwort u. Maschinenwort stimmt selten überein. Daher meist Maschinenwort kürzer als Datenwort. Aus einem Datenwort werden entsprechende Anzahlen von Maschinenworten gebildet

Wortlänge: Je nach Anwendungsgebiet sind gewisse Mindestwortlängen erforderlich. Es gibt Mikroprozessoren mit 4, 8, 12 und 16 Bit auf einem Substrat. 4 bis 32 Bit lassen sich auch mit Prozessorelementen zusammenstellen. Man unterscheidet feste und variable Wortlängen. Feste Wortlängen: Anzahl der Bits, Zeichen oder Bytes je Wort für die Operanden aller Befehle vorgegen. Variable Wortlänge: Die Wortlänge kann von Befehl zu Befehl unterschiedlich sein.

XCHG: **Ex**change the **c**ontest of register **H** and **G**, vertausche die Inhalte der Register H und G

Zeitmultiplex: Übertragung verschiedener Signale hintereinander über eine Leitung

Zentraleinheit: besteht aus Leitwerk, Rechenwerk, Arbeitsspeicher und Register

Zero: Null

Zugriff: Zugang z.B. zu einer Speicherstelle

Zyklus: Eine Anzahl von Schritten, die wiederholt werden und im Ablauf gewisse Ähnlichkeiten aufweisen.

Sachverzeichnis



Was halten Sie von diesem Buch?
Das möchten wir gerne wissen.
Setzen Sie ein paar Stichworte auf. Das genügt.
Vielen Dank sagt Ihnen der

Franzis-Verlag



Kundeninformationen

SP KL

Urteil über ______________________

______________________ (Buchtitel)

Fachgebiete, die mich besonders interessieren:

1. Hobby-Elektronik ☐
2. Elektroakustik ☐
3. Radio + TV-Elektronik ☐
4. Industrie-Elektronik ☐
5. Elektronische Meßtechnik ☐
6. Informatik (EDV) ☐
7. Amateurfunk ☐

Auf dieses Buch wurde ich aufmerksam durch:

- Prospekt ☐
- Besprechung ☐
- Schaufenster ☐
- Anzeige ☐
- Empfehlung ☐

____
Vorname/Name

____
Beruf

____
Straße

____
Postleitzahl/Ort

Hiermit bestelle ich

aus dem Franzis-Verlag, München
(Vervollständigung der ISBN genügt):

ISBN
3-7723-

kostenlose Probehefte:

FUNKSCHAU ☐

ELEKTRONIK ☐

ELO ☐

Verlagsverzeichnis ☐

Lieferung durch die Buchhandlung:

Wenn keine Firma eingesetzt, bitte senden an ▶

Bitte als
Postkarte
frankieren

Franzis-Verlag

Werbe- und Vertriebsabteilung
Bücher

Postfach 37 01 20
8000 **München 37**

93 ● ● Schaltungen für die Modellfernsteuerung (Bruß). 7. Aufl.

94 ● ● Fliegen mit Modellhubschraubern (Snitjer)

96 ● Abkürzungen aus der Elektronik (Freyer)

98 ● Jedermann-Funk (Kriebel/Rockrohr)

99 ● ● ● Wie arbeite ich mit dem Elektronenstrahl-Oszillografen? (Sutaner/Wißler). 9. Aufl.

100 ● ● Daten- und Tabellensammlung für Radiopraktiker und Elektroniker (Mende). 4. Aufl.

101 ● ● ● Elektronische Orgeln und ihr Selbstbau (Böhm). 6. Aufl.

103 ● ● Die Wobbelsender (Sutaner). 4. Aufl.

104 ● ● Transistorsender für die Modellfernsteuerung (Bruß). 6. Aufl.

105 ● ● ● Lautsprecher und Lautsprechergehäuse für HiFi (Klinger). 8. Aufl.

106 ● ● ● Netztransformatoren und Drosseln (Klein). 4. Aufl.

108 ● ● Amateurfunk-Superhets (Gerzelka). 3. Aufl.

109 ● ● ● Transistor-Amateurfunkgeräte für das 2-m-Band (Reithofer). 4. Aufl.

111 ● ● Zeigerinstrumente (Köhler). 4. Aufl.

114 ● ● Halbleiter-Experimente (Kleemann). 3. Aufl.

119 ● ● ● Gedruckte Schaltungen (Sutaner/Wißler). 4. Aufl.

126 ● ● ● Betriebstechnik des Amateurfunks (Henske). 3. Aufl.

128 ● ● ● Meßsender, Frequenzmesser und Multivibratoren (Sutaner). 2. Aufl.

134 ● ● Kleines Halbleiter-ABC (Büscher, Wiegelmann). 3. Aufl.

136 ● Transistorisierte Netzgeräte (Strobel). 4. Aufl.

147 ● ● ● Fernseh-Service leicht gemacht (Lummer). 3. Aufl.

157 ● ● Meßgeräte und Meßverfahren für den Funkamateur (Link). 2. Auflage.

160 ● ● ● Relais (Köhler).

166 ● ● Elektronik im Kraftfahrzeug (Stieber/Wilk). 2. Aufl.

168 ● ● Vademekum für den Funkamateur KW und UKW (Diefenbach/Geyrhalter). 5. Aufl.

171 ● ● ● Halbleiter-Schaltungstechnik einfach dargestellt (Benda). 3. Aufl.

174 ● ● Amateurfunkgeräte für das 70-cm-Band (Reithofer). 2. Aufl.

176 ● ● Integrierte Schaltungen für den Funkamateur (Birchel). 2. Aufl.

178 ● ● ● ● Schirmbilddiagnose und Messungen am Farbfernsehempfänger (Bochum/Dögl).

179 ● ● Digitale Elektronik für Anfänger (Kleemann). 3. Aufl.

180 ● ● ● KW- und UKW-Sender für den Funkamateur (Diefenbach). 2. Aufl.

300 ● ● Kfz-Elektronik im Selbstbau (Jansen). 2. Aufl.

301 ● ● ● Nf-Elektronik (Sabrowsky). 4. Aufl.

302 ● ● ● Electronic-Pianos und Synthesizer (Tünker).

304 ● ● ● Transistor-Schaltverstärker (Sabrowsky). 2. Aufl.

319 ● ● ● Der leichte Start zum Funkfernsteuern (Sabrowsky). 3. Aufl.

322 ● ● ● Impulstechnik für jedermann (Sabrowsky).

331 ● ● ● Verstärkerbau mit integrierten Schaltungen (Wirsum). 2. Aufl.

335 ● ● ● Mischpulte und Mischpultmodule (Wirsum). 3. Aufl.

336 ● ● ● Musikelektronik (Tünker). 3. Aufl.

337 ● ● ● Elektronische Hilfsmittel für Film und Foto (Horst).

338 ● ● Experimente mit IS (Kleemann). 2. Auflage.

339 ● ● ● HiFi-Boxen im Selbstbau (Sabrowsky). 3. Auflage.

340 ● ● ● Vom Flip-Flop zur Quarzuhr (Pelka). 2. Aufl.

341 ● ● Modelleisenbahn-Elektronik im Selbstbau (Jungmann). 2. Aufl.

342 ● ● ● Elektronik in der Popmusik (Goddijn).

Pelka, Was ist ein Mikroprozessor?

3. Auflage

82

Doppelband



Ja, was ist nun ein Mikroprozessor? Seit wann gibt es ihn überhaupt? Wie ist er entstanden? Hat ihn jemand erfunden? Wie ist er aufgebaut? Welche Technologien werden dabei verwendet? Können einzelne Bausteine ausgetauscht werden? Ist er eine Fortentwicklung der Mikrocomputersysteme? Wie ist seine Arbeitsweise? Ist er programmierbar? Wie weit? Welche Programmiersprachen versteht er? Können seine Programme geändert werden? Kann das jedermann? Ist die Programmierung stromausfallsicher?

Was leistet ein Mikroprozessor? Wie und wo kann er angewandt werden? Wird er den Taschencomputer ersetzen? Gibt es bereits Standardtypen? Inwieweit unterscheiden sie sich voneinander? Welches sind die Auswahlkriterien für die Anwendung?

Fragen über Fragen! Die Antworten gibt dieser Band.

Der Autor: Horst Pelka ist einer der wenigen Universal-Elektroniker. Dazu kann er noch didaktisch hervorragend schreiben (RPB 38, RPB 340). Hauptberuflich arbeitet er seit Jahren in einem großen Elektronik-Konzern.

FRANZIS



ISBN 3-7723-0823-6